1987

Investigation of two ferromagnetic damping materials in conjunction with initial development of a signal analyzer interface program.

Patch, Gregory Richard.; Perkins, Jeffery Arthur

http://hdl.handle.net/10945/22831

NAVAL POSTGRADUATE SCHOOL

Monterey, California

INVESTIGATION OF TWO FERROMAGNETIC DAMPING MATERIALS IN CONJUNCTION WITH INITIAL DEVELOPMENT OF A SIGNAL ANALYZER INTERFACE PROGRAM

by

Gregory Richard Patch

September 1987

Thesis Advisor: Jeff Perkins

Approved for public release; distribution is unlimited

UNCLASSIFIED

JRITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION :lassified	1b RESTRICTIVE MARKINGS None
SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT approved for public release; distribution is unlimited
DECLASSIFICATION/DOWNGRADING SCHEDULE	
ERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
NAME OF PERFORMING ORGANIZATION 7al Postgraduate School — 6b OFFICE SYMBOL (If applicable) Code 69	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
ADDRESS (City, State, and ZIP Code) :terey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
IAME OF FUNDING/SPONSORING ORGANIZATION — 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

ITLE (Include Security Classification)
ESTIGATION OF TWO FERROMAGNETIC DAMPING MATERIALS IN CONJUNCTION WITH
TIAL DEVELOPMENT OF A SIGNAL ANALYZER INTERFACE PROGRAM

ERSONAL AUTHOR(S)
'CH, Gregory R.

TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year Month Day)	15 PAGE COUNT
ter's Thesis	FROM ______ TO ______	1987 September	151

UPPLEMENTARY NOTATION

COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
ELD	GROUP	SUB-GROUP	HIGH DAMPING: Fe-Cr-Mo; Fe-Cr-Al; COMPUTER ANALYSIS; INTERFACE PROGRAM

BSTRACT (Continue on reverse if necessary and identify by block number)

romising method for the attenuation of shipboard vibration and noise lizes constituent materials whose composition and microstructure come to absorb energy internally and dissipate it. However, the alloy ign of damping materials which possess both significant energy-absorpn capacities and also necessary strength levels is difficult. Pretly, there are several problems associated with designing a material h high damping characteristics. One of these involves the development efficient, reliable and reproducable methods for measurement of mater- damping capacities. In this study, the primary area of interest conned the development of microcomputer analysis techniques to study the ration damping response of two iron-chromium (Fe-Cr)-based alloys. The sent research utilized a Zenith Corporation Z-150 microcomputer to com-

STRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
IAME OF RESPONSIBLE INDIVIDUAL fessor J. Perkins	22b TELEPHONE (Include Area Code) (408) 646-2216 — 22c OFFICE SYMBOL Code 69Ps

ORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

pose programming that captures, stores, and analyzes the damping data produced by various Fe-Cr-based alloy specimens. The computer programming developed in the present research enables an interface of the Zenith Z-150 computer with a Scientific Atlanta SD380Z Signal Analyzer. The programs written will: (a) store analyzer screen displays on computer disc media, (b) facilitate damping measurements, (c) produce graphic displays of alloy damping characteristics, (d) calculate damping capacities, (e) operate with commercially available hardware and software, (f) provide a programming tool for subsequent researchers to promote further development of this technique.

Approved for public release; distribution is unlimited

Investigation of Two Ferromagnetic Damping Materials in Conjunction with Initial Development of A Signal Analyzer Interface Program

by

Gregory Richard Patch
Commander, United States Navy
B. S., United States Naval Academy, 1970

Submitted in partial fulfillment of the requirements for the degree to

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1987

DEDICATION

For my loving wife, Marci, and lovely daughter, Felicia.

ABSTRACT

A promising method for the attenuation of shipboard vibration and noise utilizes constituent materials whose composition and microstructure combine to absorb energy internally and dissipate it. However, the alloy design of damping materials which possess both significant energy-absorption capacities and also necessary strength levels is difficult. Presently, there are several problems associated with designing a material with high damping characteristics. One of these involes the development of efficient, reliable and reproducable methods for measurement of material damping capacities. In this study, the primary area of interest concerned the development of microcomputer analysis techniques to study the vibration damping response of two iron-chromium (Fe-Cr)-based alloys. The present research utilized a Zenith Corporation Z-150 microcomputer to compose programming that captures, stores and analyzes the damping data produced by various Fe-Cr-based alloy specimens. The computer programming developed in the present research enables an interface of the Zenith Z-150 computer with a Scientific Atlanta SD380Z Signal Analyzer. The programs written will: (a) store analyzer screen displays on computer disc media, (b) facilitate damping measurements, (c) produce graphic displays of alloy damping characteristics, (d) calculate damping capacities, (e) operate with commercially available hardware and software, (f) provide a programming tool for subsequent researchers to promote further development of this technique.

TABLE OF CONTENTS

I. INTRODUCTION

A. GENERAL

Audio noise reduction and vibration control of shipboard propulsion and auxiliary machinery hold obvious benefits for the Navy and are currently under investigation for application in United States Navy submarine and surface ship design. (Ref. 1: p.12) (Note 1) Conventional structural alloys generally do not exhibit significant damping capacities at stress amplitudes associated with machinery vibration and noise emission. Under prevailing conditions, as outlined by Schetky and Perkins (Ref. 2), there are three principle methods of vibration and noise control which are employed:

1. Isolation of the machinery source of vibration and noise from the surfaces to which it radiates energy.
2. Dissipation and attenuation of vibration energies generated by machinery through the use of absorbing pads (i.e., rubber insulation for pipe hangers, resilient motor mounts, etc.).
3. Attenuation of vibration and noise through the use of constituent materials whose composition and microstructure combine to absorb energy internally and dissipate it.

The first two methods are commonly employed and are effective, but they contribute significant additional weight and utilize valuable space, undesirable aspects for machinery applications aboard submarines and surface ships. The third method is less

widely applied, but has the potential to eliminate these problems, since a design may be possible without cumbersome attachments or supporting structures. However, the alloy design of high damping materials which fulfill the desired energy-absorption functions in lieu of machine parts and also satisfy the necessary corrosion properties and strength requirements of constituent materials is very difficult. (Ref 1: p. 13)

Presently, there are several problems associated with designing a material with high damping characteristics (Ref. 3: p.1).

1. Reliable and reproducable methods for accurate measurement of the energy conversion process occurring within the material are still to be developed.

2. Better understanding is required of the fundamental processes involved in the energy transfer occurring within the material (i.e., the damping mechanisms) and how these processes relate to the microstructure.

3. Preservation of damping capacities subsequent to mechanical forming is required.

4. Resistance to corrosion in a marine environment requires further investigation, particularly for the Fe-Cr alloys (Ref. 4).

Current research at the U.S. Naval Postgraduate School on materials with high internal damping mechanisms is directed toward determination of the specific damping capacities of existing commercial alloys and the relationships between the mechanisms of damping, microstructure and physical properties. Ultimately, this research should lead to the development of

alloys with optimum damping, structural forming and strength characteristics.

B. BACKGROUND

All materials display some measure of energy absorption or dissipation phenomena. However, most metallic alloys exhibit poor damping capacities at stress amplitudes associated with machinery vibration and noise emission. Specific damping capacities for common structural alloys, such as steels, brasses and aluminum alloys, are less than 1% See Table 1. (Ref. 1: p. 16). Cast materials, such as gray cast iron, typically have higher damping capacities than wrought products. The primary feature of gray cast iron which damps vibration is the large, flake-like particles of graphite in the microstructure. These large graphite flakes are effective in absorbing energy, resulting in specific damping capacities ranging from 5% to 10% (Ref. 2: p.203). Other types of cast iron, containing rounded graphite particles within the microstructure, have lower damping capacities.

TABLE 1

DAMPING CHARACTERISTICS OF SELECTED METALS AT ROOM TEMPERATURE (REF. 2: p.16)

METAL	SDC (%)	YIELD STRENGTH (10^3 PSI)	DENSITY (gm/cm3)
Magnesium (wrought)	49	26	1.74
Cu-Mn alloys (INCRAMUTE, SONOSTON)	40	45	7.5
Ni-Ti alloy	40	25	6.45
Fe-Cr-Al alloy	40	40	7.4
High-C gray iron	19	25	7.7
Nickel (pure)	18	9	8.9
Iron (pure)	16	10	7.86
Martensitic stainless steel	8	85	7.7
Gray cast iron	6	25	7.8
SAP (aluminum powder)	5	20	2.55
Low-carbon steel	4	50	7.86
Ferritic stainless steel	3	45	7.75
Malleable, nodular cast irons	2	50	7.8
Medium-carbon steel	1	60	7.866
Austenitic stainless steel	1	35	7.8
1100 Aluminum	0.3	5	2.71
Aluminum alloy 2024-T4	<0.2	47	2.77
Nickel-base superalloys	<0.2	Range	8.5
Titanium alloys	<0.2	Range	4.5
Brasses, bronzes	<0.2	Range	8.5

In recent years, several new alloys have been developed which exhibit significantly higher specific damping capacities (in excess of 40%), including Cu-Mn-based alloys (SONOSTON, INCRAMUTE), Ni-Ti alloys (NITINOL), and Fe-Cr-based alloys (SILENTALLOY, VACROSIL, GENTALOY, etc.). Damping inherent to these high damping alloys, as for any such material, originates within the microstructure. The microstructure of a material determines the mechanisms responsible for internal damping of external excitations. Some variables of the imposed vibration, such as frequency, strain amplitude and number of cycles, can of course also

influence the vibration damping response of a particular alloy. (Ref. 5) The character of the imposed excitation can produce a variety of microstructural responses, which in general result in damping due to irreversibilities incurred when a material is subjected to an alternating stress. The absorption of energy via "internal friction" is due to these irreversibilities, whether the "friction" is due to ferromagnetic domain walls or intercrystalline boundaries. The internal friction results in the energy loss per cycle which is referred to as damping. The energy losses which characterize these irreversibilities may be evident in stress-strain hysteresis, representing ferromagnetic, mechanical, and/or thermal losses. The amount of energy loss is closely tied to material microstructure as a function of:

* the constituent elements within an alloy system
* thermomechanical history
* environmental temperature and whether it is above or below a "threshold" temperature
* the imposed stress level, perhaps with respect to a critical stress level

Under cyclic (or periodic) stress, several microstructural damping mechanisms are possible, depending upon the material. Among them are:

* dislocation damping (Refs. 5, 6)
* interphase boundary damping (Ref. 5)
* phase change effects (Ref. 5)

Previous research on "quiet alloys" at the U.S. Naval Postgraduate School had been concerned mainly with nonferrous alloys, specifically the Cu-Mn based alloys SONOSTON (Ref. 7) and INCRAMUTE (Ref. 8). Damping mechanisms for these alloys apparently involve the interaction of cyclic stress with twinned and "tweed" microstructures (interphase boundary damping). (Ref. 1)

In this study, as in O'Toole's (Ref. 1) and Ferguson's, (Ref. 8a), the area of interest concerns itself with iron-chromium (Fe-Cr)-based alloys, and specifically an iron-chromium-molybdenum (Fe-Cr-Mo) alloy and an iron-chromium-aluminum (Fe-Cr-Al) alloy, the compositions for which are given in Appendix A, and which were were verified by two independent analysis reports. Previous research by Cochardt (Ref. 9), de Batist (Ref. 5), Schilling and Houze (Ref. 10), Willertz (Ref. 11), Suzuki, et al. (Ref. 12), Masumoto, et al. (Refs. 13-15), Schneider, et al. (Ref. 16), and Kasper (Ref. 17) has established that the damping mechanism for this alloy group results from the ferromagnetic properties of the material. According to de Batist (Ref. 5: p.43), the damping mechanism is domain boundary damping due to the magnetomechanical interaction phenomena which naturally occurs in ferromagnetic materials.

High vibration damping has been reported for several ferrous alloys, particularly for the binary systems Fe-Cr (12 - 16 wt. % Cr) (Ref. 13) and Fe-Mo (2 - 5 wt. % Mo) (Ref. 14). It has also been reported for the ternary

systems Fe-Cr-Mo (Refs. 12, 15) and Fe-Cr-Al (Refs. 18-21). A recently introduced high damping Fe-Cr-based alloy has been placed on the commercial market by Vacuumschmelze, G.M.B.H. (VAC), of Hanau, West Germany, with the registered trade name of VACROSIL-010. This material is available in two similar compositions, Fe-Cr-Al and Fe-Cr-Mo, the latter being called a corrosion resistant grade (Ref. 17). The corrosion resistant version of VACROSIL-010 (Fe-Cr-Mo) is of special interest to the Navy. For direct information regarding its corrosion resistance in seawater, reference should be made to Escue (Ref. 4).

C. OBJECTIVES

This study used a recently developed method for damping measurement; the single cantilever beam resonance dwell technique, discussed in greater detail later in this introduction. The resonant dwell technique employs forced vibration to determine the loss factor and damping coefficient of a simple cantilever beam by measuring its response to excitation at a modal frequency. (Ref. 26) This research augmented the resonant dwell measurement approach by exploring the utilization of a microcomputer to process the damping data produced by the vibrating catilever beam.

The microcomputer currently has sufficient memory capacity to permit the processing of large programs and volumes of data. That advantage is useful to analyze the entire damping response of a vibrating cantilever beam at a resonant node. Although the assignment of Specific Damping

Capacities (SDC) and/or damping coefficients partially characterize a mechanical energy absorption material, individually those and similar numerical parameters may be insufficient to fully describe the damping response. Today, affordable equipment such as the microcomputer is available which permits broader characterization and analysis of the mechanical response of a material.

This research utilized a Zenith Z-150 microcomputer in conjunction with a Scientific Atlanta SD380Z Signal Analyzer to capture, store and analyze the dampening data produced by various specimens. Most importantly, a BASIC microcomputer program (Appendix E) was developed which accomplished the following objectives.

* It provides a programming tool for further development of simple programs for compilation and analysis of materials damping data.
* It provides limited automation that simplifies and facilitates the laboratory data collection and analysis process.
* It illustrates the potential of this approach.
* It operates with commercially available hardware and software.

For the actual vibration experiments, this research considered Fe-Cr-Al and Fe-Cr-Mo alloys of similar compositions (Appendix A) to the VACROSIL alloys with the following objectives:

* To further determine the relationship between the room temperature damping properties of Fe-Cr-Al and Fe-Cr-Mo to applied strain (or stress) under random vibration conditions for various heat treatment histories; one hour annealing at various temperatures between 950 C and 1100 C followed by subsequent furnace cooling. This narrow region of heat treatments was

chosen because prior work (O'Toole, Ref. 1) indicated that such heat treatments produced the greatest damping capacities.

* To produce graphic displays of alloy dampening characteristics versus sample heat treatment histories.

* To attempt to image the ferromagnetic domains thought responsible for observed damping (Note 2).

* To document the microcomputer analysis methods which were developed to enable further development of this technique by subsequent researchers.

D. MACROSTRUCTURAL DAMPING, MATHEMATICS & MEASUREMENT TECHNIQUES

1. Mathematics for Damping Expressions

The ability of a material to absorb vibrational energy and convert such energy to other forms, such as heat or mechanical friction, is defined as the damping capacity of the material. Specific damping capacity is the fraction of input vibrational energy or amplitude absorbed during one cycle of vibration. There are several methods to characterize the vibration damping of a material (Ref.1: p.18). Some of the more frequently employed methods are:

a. Logarithmic Decrement (delta, d)

The natural logarithm of the ratio of any two successive decaying amplitudes in time is the definition of logarithmic decrement. Free vibrations decay exponentially for a linear system. Thus, the faster the decay, the larger the decrement, indicating greater damping. (Ref. 22: p.138) See Figure 1.

$$d = \ln(a(i)/a(i+1)) = (1/n)\ln(a(0)/a(n)) \qquad (1)$$

where n = number of cycles between a(0) and a(n)

b. Quality Factor (Q)

The quality factor, often used in electronics signal analysis, is a measurement of the sharpness of a resonance peak. (Ref. 23: p.76) See Figure 2.

$$Q = wn/(w2 - w1) = 1/2 * Zeta \qquad (2)$$

where Zeta = damping factor and w2 & w1 are 3db lower than wn

c. Internal Friction (1/Q)

Internal friction is defined as the inverse of the quality factor. (Ref. 5: p.41)

$$1/Q = (w2 - w1)/wn = 2 * Zeta \qquad (3)$$

This expression is also known as the loss factor.

d. Normalized Bandwidth (Figure 2)

From the half power method, damping can be measured in terms of the frequency width (w2 - w1) of the peak at the resonant frequency (wn) at points on the curve corresponding to 0.707 (1/square root of 2) of the peak amplitude. This calculation corresponds to the points on the resonant peak at which the stored energy is half its maximum value at the resonant frequency and is normalized by that frequency. (Ref. 21: p.76)

e. Specific Damping Capacity (SDC)

SDC is the percent of strain energy dissipated during a stress cycle for a material undergoing oscillating excitation. (Ref. 24: p.444).

$$SDC = [a(i+1)^2 - a(i)^2]/a(i)^2 \qquad (4)$$

If [a(i+1) - a(i)] is small, SDC can be approximated by (Ref. 24: p.444):

$$SDC = 2[a(i+1) - a(i)]/a(i) \quad (5)$$

f. Phase angle (@)

The angle by which strain lags behind stress in cyclic or periodic loading is defined as the phase angle. (Ref. 24: p.445)

$$\tan @ = (1/pi)*\ln[a(i)/a(i+1)] = d/pi \quad (6)$$

where pi = 3.1415927... = circumference/diameter

For small values of damping (zeta < 0.5), the following relations hold: (Ref. 8: p.18-19)

$$\tan @ = d/pi = 1/Q = 2*zeta$$

$$SDC(\%) = 200*pi/Q = 200 * pi * [(w2-w1)/wn] \quad (7)$$

Specific Damping Capacity (SDC) was the parameter primarily used for damping measurements in this report.

g. Resonant Frequency (w(n))

Resonant frequency is the natural frequency of a cantilever beam corresponding to the nth mode (Ref. 23: p.464):

$$w(n) = [Cn^2][E*I/m*l^4]^1/2 \quad (8)$$

where E -- Young's Modulus of Elasticity

I -- Moment of Inertia

m -- beam mass per unit length

l -- vibrating length of beam

Cn -- function of the mode of vibration of the beam; values for several different modes (n) are given below for a clamped cantilever beam (Ref. 23: p. 466)

n	Cn	Cn^2
1	1.8751	3.5160
2	4.6941	22.0345
3	7.8548	61.6972

2. Damping Measurement Techniques

Damping measurements on Fe-Cr alloys have been performed in the past primarily using either an inverted torsion pendulum apparatus (Refs. 13-15, 19) under free decay conditions, or via forced vibration (Ref. 11). Another method which has been utilized employs the cantilever beam (Ref. 12, 16) wherein damping is also determined from the decay of free oscillations.

This study used a method that has begun to receive increasing attention, the single cantilever beam resonance dwell technique. This method, developed by Bolt, Beranek and Newman, Inc. (Ref. 25) can be used to determine the stress and frequency dependence of material damping over a 25 - 2000 Hz frequency range. The resonance dwell technique is an induced vibration method for determining the loss factor of a simple structural element by measuring its response to excitation at a modal frequency. This technique was demonstrated for vibration damping measurement by Kaufman, Kulikn, and Neshe (Ref. 26) for NiTi and CuAlNi. To summarize this approach, a cantilever beam is clamped to a bar; the bar in turn is connected at one end to an electromagnetic shaker and to a heavy base at the other. An accelerometer is mounted at the root of the sample. The input excitation signal is measured by this accelerometer.

The cantilever beam (specimen) response is measured by a second accelerometer at the free (unsupported) end of the beam. The fundamental (first modal) natural frequency "f" of a cantilever beam of length "l" (inches) and thickness "t" (inches) is: (Ref. 25: p.6)

f = (t/2 * pi) * (Cn/l)^2 * (32E/Ro)^1/2 (cycles/sec) (9)

where Cn is from (Eq. 8), Ro is the density of the sample in (lbs/in^3) and E is the dynamic Young's Modulus. "l" is the vibrating length of the beam, while the width (an independent factor) is 0.5 inches, conforming to the width of the bar to which the specimens are clamped. See Figure 3.

The positions of the first three nodal points (n = 1,2,3) from the root of the beam, located as a fraction of the entire beam length, can be shown to be: for n = 1, l = 0.0; for n = 2, l = .53; for n = 3, l = 0.31 to 0.71.

3. Microcomputer Utilization

The practice of using two accelerometers was initially developed by Professor Y.S. Shin of the Naval Postgraduate School. In the recent work, signals produced by the input accelerometer (at the root of the specimen) and the output accelerometer (at the tip) were processed for this study by a Scientific Atlanta spectrum analyzer to produce the frequency response of the vibrating beam at a resonant mode. This method has been tested and compared to a forced torsion pendulum device for the measurement of damping in SONOSTON by Dew (Ref. 7) and further tested by

Reskusich (Ref. 8) for INCRAMUTE, and by Cronauer (Ref. 27) for Ti-Ni and Fe-Cr-Mo. Since the signal analyzer has a built-in computer interface bus and programable memory, a Zenith Z-150 microcomputer was used to operate the analyzer much of the time. Using the computer permitted: (1) storing analyzer set-up configurations, and using such set-up files to configure the analyzer much faster; (2) capturing dampening data in hard disc files and analyzing it at will; (3) combining the graphic displays relating damping characteristics of several samples on one graph; (4) writing the final report, etc.

E. MICROSTRUCTURAL DAMPING MECHANISMS OF FERROMAGNETIC ALLOYS

As discussed previously, the primary microstructural mechanisms which contribute to high damping are:

* dislocation damping
* interphase boundary damping
* phase change effects

Damping in ferromagnetic materials is generally the result of two damping mechanisms. The damping caused by the magneto-mechanical hysteresis effect is the primary mode of damping. This damping mechanism is described by de Batist (Ref. 5, p.43) as a form of interphase boundary damping. A secondary mode of damping in these materials is due to the interaction of dislocations within the material (Ref. 27A). This paper is concerned with the first and primary damping mode.

Ferromagnetic materials manifest magnetic domains, which are more or less randomly oriented in an unmagnetized material. Upon the application of a magnetic field, or a unidirectional tensile (or compressive) stress, these domains tend to align with the direction of the tensile strain. Any subsequent movement of these domains produces an irreversible (but restorable) change in the dimensions of the material called "magnetostriction." When a stress-strain curve for an unmagnetized ferromagnetic material is plotted, more strain is measured than is postulated by Hooke's Law. Upon gradual removal of the load, the elastic strain reverts to zero (assuming negligible plastic deformation), but the magnetostrictive strain remains nearly constant, (i.e.), the unloading curve follows a hysteresis loop. The greater the area of the hysteresis loop, the larger the damping capacity of the material, from which it may be inferred that the damping capacities of high-strength ferromagnetic alloys are functions of magnetostriction and stress. (Ref. 1: p.26)

Considering microstructure, and summarizing, damping properties of the material are related to the movement of domain boundaries upon the application of stress. In their work with grain-oriented 3% Si-Fe, Schilling and Houze (Ref. 10) outlined their theory regarding magnetic domains as follows:

a. Ferromagnetic domains are small magnetically ordered crystal regions within which magnetization is equal to the saturation magnetism. Therefore, the net magnetization is

a vector sum of the magnetization for all of the domains.

b. Upon the application of a magnetic field or external stress to a magnetic material, (orientation) changes in the domain structure occur which produce changes in the overall specimen magnetization, as well as the specimen dimensions, (i.e.) "magnetostriction."

c. Response to changes in an external field may be manifested in one of two primary manners. Either magnetization within each domain may coherently rotate to a direction parallel to the applied field, or the boundary between two domains may move; in the latter case, the changing magnetization is entirely localized at the domain boundary.

In materials such as cast 3% Si-Fe, magnetic domain misalignment is predominant; therefore, domain rearrangements occur by the movement of domain boundaries or walls between domains. These are called Bloch walls (Ref. 28: p.613) and are considered to be about 1000 angstroms thick. Bloch walls function in a manner similar to grain boundaries. They are narrow zones in which the magnetic moment vector changes from one domain to the next. Such domain boundaries have been imaged using an electron microscope by H.W. Fuller and M.E. Hale (Ref. 29), S. Amelinckx (Ref. 30), J. Silcox, E. Fuchs and others.

Stressing a ferromagnetic material acts to align the magnetic domains in the direction of the stress. Under stress (or a weak magnetic field), domains aligned with the applied field tend to grow at the expense of neighboring domains whose directions are less favorably oriented. As the applied stress (field) becomes stronger, it can also produce a rotation of the magnetic moment vector within

domains toward the direction of the applied stress (field) (Ref. 24: p.133). This domain movement results in an irreversible change in the material called magnetostriction. When energy imparted to the system in the form of mechanical vibrations produces this transformation, the resulting attenuation of applied vibrational force constitutes damping, and is in fact a relatively potent damping mechanism. (Ref. 1: p.28)

Another important point concerns the magnetic transition or Curie temperature of the material. When the temperature of a ferromagnetic material is increased, the added thermal energy reduces its degree of magnetization by permitting random domain reorientation. Heating above the Curie temperature for a short period completely transforms the material to a non-magnetic (paramagnetic) state in which the domains are randomly oriented throughout its microstructure. (Ref. 1: p.27)

Crystalline materials, whether ferromagnetic or not, exhibit effects in response to periodic stress due to dislocation damping. Under an applied stress, dislocations move in an oscillatory manner and energy is absorbed by the material. Of course, if the applied stress is high enough, the material will react plastically, and undergo an irreversible shape change.

F. MATHEMATICAL RELATIONS FOR DAMPING MECHANISMS OF FERROMAGNETIC ALLOYS

The relationship between parameters of macrostructural damping in ferromagnetic materials is provided by Cochardt (Ref. 9: p.197-199) for damping capacity, magnetostriction, critical stress, and maximum stress. The damping capacity is expressed as the logarithmic decrement (delta or "d"):

$$d = (1/2\ (Uv/U) \qquad (10)$$

where U = mean elastic energy of the specimen.

$$U = (1/V)*(1/2)*\text{integral}\{\ (\text{sigma}^2)/E\ dV\} \qquad (11)$$

where V is volume; sigma is normal stress; and E is Young's modulus.

Uv is the energy dissipated in the entire specimen per cycle per unit volume (Ref. 9: p.197)

$$Uv = (1/V) * \text{integral}\{\ dU\ dV\ \} \qquad (12)$$

where dU is the elemental energy loss per cycle and unit volume at the volume element dV. dU represents the area of the hysteresis loop due to the magneto-mechanical effect. For small stresses, the area of the hysteresis loop is described by: (Ref. 9: p.198)

$$dU = D * (\text{sigma}^2) \qquad (13)$$

where D is a constant according to Rayleigh's law. For stresses larger than a maximum or critical (sigma-c) stress, beyond which the area of the hysteresis loop remains constant, dU is constant and can be written as: (Ref. 9: p.198)

$$dU = K * \text{lambda} * \text{sigma-c} \qquad (14)$$

where K = 4 for an ideal parallelogram-shaped loop

```
      K = 1 for most other cases
     lambda = saturation magnetostriction in  the
              easy direction of magnetization
```

The assumption is made that Rayleigh's Law is valid up to this critical stress. Therefore, (Ref. 9: p.198)

```
    dU = D(sigma^2)            0 < sigma < sigma-c
    dU = K lambda sigma-c      sigma-c < sigma < sigma-m

---> D  = K lambda/(sigma-c)^2  {sigma-m is maximum normal
                                 stress in a cantilever beam}
```

Substituting the above relation into equation 12 and replacing dV by (dV/dsigma)dsigma, the logarithmic decrement becomes: (Ref. 9: p.198)

```
d = (1/V)*(K lambda/2U)*integral{ sigma^2/(sigma-c)^2
                  0 --> sigma-c      * (dV/dsigma)dsigma }

     +     integral{ sigma-c(dV/dsigma)dsigma }
           sigma-c --> sigma-m                             (15)
```

As previously defined, Q^-1 = d/pi.

Therefore, equation 15 can be rewritten as...

```
Q^-1 = (1/piV)*(K lambda/2U)*integral{ sigma^2/(sigma-c)^2
                     0 --> sigma-c     * (dV/dsigma)dsigma }

    + (1/pi) * integral{ sigma-c(dV/dsigma)dsigma }
               sigma-c --> sigma-m                         (16)
```

It should be noted that dV/dsigma is the stress distribution function. This can be evaluated in terms of the stress conditions of a cantilever beam:

```
   sigma = M z / I                                         (17)

where  M = bending moment
       z = distance from the neutral axis of the
           cantilever beam
       I = moment of inertia of beam rectangular
                  cross section
```

For a cantilever beam, Cochardt continues this derivation and defines the logarithmic decrement as: (Ref. 9: p.199)

d = 9K*lambda*E*(sigma-c/(sigma-m)^2 * (1 - sigma-c/sigma-m

* ((3/4)ln(sigma-m/sigma-c) + 15/16))

for sigma-c < sigma-m (18)

This equation provides an analytical expression for damping which relates several of the pertinent variables. It is apparent that after sigma-c is reached, further stress tends to reduce the resultant damping, as the squared term predominates. Further, a larger elastic modulus promotes greater damping in such materials.

Damping associated with a ferromagnetic material thus reaches a maximum value at a point of critical stress. Beyond this point, gradually decreasing values of damping are recorded with increasing stress. This is attributable to a saturation condition for damping wherein the existing domains cannot grow or move any further. In addition, Degauque, Astie and Kubin (Ref. 27A) report from their experiments with high purity iron that the interaction between 90 degree magnetic domain walls and dislocation tangles appears to be a major obstruction to the motion of magnetic domain walls. Single defects like isolated dislocations can interfere with small displacements of magnetic domain walls but they can not substantially oppose the large scale movement of these walls in the vicinity of maximum damping. Therefore, an increase in dislocation density produces a decrease in the intensity of maximum damping. (Ref. 1: p.31-32)

G. METALLURGY OF THE IRON-CHROMIUM ALLOY SYSTEM

1. Physical properties of the Fe-Cr Alloys

The composition range of Fe-Cr binary alloys which are of interest as high damping ferromagnetic alloys is similar to that for ferritic stainless steels, one of the three main classes of stainless steels (the other two being austenitic and martensitic alloys). Ferritic stainless steels are iron based alloys with a chromium content ranging between 12 and 30 weight percent. The use of ferritic stainless steels has been much more restricted than austenitic stainless steels because ferritic steels are susceptible to embrittlement, are notch sensitive, and exhibit poor weldability; factors which contribute to poor fabricability. However, advantages of ferritic stainless steels include high resistance to stress-corrosion cracking, and good to excellent corrosion and oxidation resistance. Ferritic stainless steels are also known to have excellent damping properties. (Refs. 11 & 31)

Ferritic stainless steels are structurally quite simple. At room temperature, the Fe-Cr (alpha) solid solution has a body-centered crystal (bcc) structure. These alloys contain very little dissolved carbon, the majority of which appears in the form of finely divided chromium carbide precipitates. (Ref. 1: p.33)

The Fe-Cr binary phase diagram (Figure 4) exhibits a great deal of activity in the 11% to 12% Cr content region. As outlined by Peckner and Bernstein (Ref. 32: p.5-2 - 5-3),

the following relations and potential transformations exist. (Ref. 1: p.33-34)

* As a member of a group of elements described as ferritic stabilizers, chromium extends the (alpha) phase field while narrowing and suppressing the (gamma) face-centered (fcc) phase field. As evidenced in Figure 4, this creates a "gamma loop" extending in temperature range from 850 degrees Centigrade (C) to 1400 degrees C and from zero to about 12.5 weight percent chromium.

* Whereas the transformation from alpha to gamma phase occurs in pure iron at about 910C, at an 8% concentration, chromium depresses the transition temperature to about 850C. Upon further addition of chromium, the transition temperature rapidly increases to about 1000C as the chromium content reaches 12% to 13%.

* Whereas in pure iron the inverse transformation from gamma to alpha occurs at about 1400C, this reaction is depressed to about 1000C in the 12% to 13% Cr range. Also at this point in the phase diagram (1000C, 12% to 13% Cr), the upper and lower temperature alpha:gamma curves join to close off and form the gamma loop. Beyond 12% to 13% Cr, transformation to gamma is no longer possible and an alloy would remain ferritic (bcc) over the entire range from room temperature to melting.

* Between the extensive alpha phase field and the gamma loop, there is a relatively narrow transition band where the alloy can have both alpha and gamma phases. Because of the narrow extent of this two phase region, depending on the annealing temperature, alloy composition and quench rate, a two phase composition may or may not be retained upon cooling to room temperature.

The defining parameters of the gamma loop have been established for the Fe-Cr binary system through the work of Baerlecken, Fisher, and Lorentz (Ref. 32: p. 5-3). Using magnetic measurements at elevated temperatures, the lowest point in the gamma loop was identified at 840C and 6.5% Cr. The greatest width of the alpha and gamma phase field occurred at 1075C and reached to about 11.5% Cr. Variations in the extent of the gamma loop were found to be very much a

function of the addition of austenizing elements, particularly carbon and nitrogen. Increasing levels of these interstitial elements causes the gamma loop to extend to higher chromium levels See Figure 5.

Another effect of carbon is that because of its low solubility in the alpha phase, excess carbon is rejected from the solid gamma solution to form complex carbides, such as (Cr,Fe)7 C3 and (Cr,Fe)23 C6, which precipitate predominately along grain boundaries. (Ref. 1: p.35) These grain boundary precipitates are a primary factor behind the lower toughness of ferritic steels.

The strengthening mechanisms normally characteristic of stainless steels do not apply to the ferritic stainless steels. Ferritic stainless steels are characterized by the absence of an alpha --> gamma transition upon heating to high temperatures. Consequently, hardening that occurs as a result of a gamma --> martensite transformation upon cooling will not normally occur. (Ref. 1: p.36)

The greatest disadvantage to the use of ferritic stainless steels has been a loss of corrosion resistance and ductility following exposure to high temperatures. After certain heat treatments, chromium precipitates out of solution as chromium-carbides along grain boundaries, thus reducing the desirable characteristics impacted by chronimum. However, the addition of molybdenum (Mo) improves the corrosion resistance of ferritic stainless steels.

Molybdenum forms benign carbide precipitates and allow Cr to remain in solution upon exposure to elevated temperatures.

2. Damping Properties of Fe-Cr Alloys

The damping properties of Fe-Cr alloys are attributable to the magneto-mechanical hysteresis mechanism associated with ferromagnetic materials. This mechanism is directly related to the physical state of the material and the associated microstructure. The following physical parameters affect the magnetic domain wall mobility of the material and subsequently its damping capacity. (Ref. 1: p.36)

a. Strain or Stress (Refs. 9,11-13,15-17,18,19,21)

The degree to which stress influences damping depends on the alloy's thermo-mechanical history. In general, damping capacity increases with applied stress or strain. Damping will reach a maximum value with stress beyond which further stress will lower damping values.

b. Cold Work (Refs. 16,17,26)

Damping capacity is strongly deteriorated by cold work. A reduction of >= 5% completely destroys the damping effect; however, it can be fully restored by a succeeding heat treatment.

c. Magnetic Field (Refs. 9,11,14-17,20,33)

At high fields, the domain walls become fixed; i.e., the damping capacity decreases and finally disappears. Therefore, these alloys should not be used in

applications where there are stray magnetic intensity fields greater than the range of 50 - 100 A/cm.

d. Magnetic (Curie) Transformation Point (Refs. 28, 32,34)

The magnetic transformation temperature, otherwise known as the Curie temperature, is the point above which iron is paramagnetic and, below which it is ferromagnetic. Paramagnetic iron is nonmagnetic (permeability = 1.00). Ferromagnetic iron is magnetic (permeability > 1.00), the magnitude varying with composition. At room temperature gamma-austenite (fcc) is nonmagnetic while alpha-ferrite (bcc) is ferromagnetic. Therefore, the magnetic composition and hence the ultimate damping capacity is affected by the degree of alpha-ferrite present in the structure.

Note 1: The basic format and text for this introduction were patterned after a thesis by John F. O'Toole, who conducted previous research regarding Fe-Cr vibration damping alloys (Ref. 1). However, this introduction significantly modifies and amplifies that original reference.

Note 2: Visiting Professor Yamashida conducted the electron microscope work to attempt imaging of the ferromagnetic domain walls. His assistance was gratiously provided for use in this report.

II. EQUIPMENT AND EXPERIMENTAL PROCEDURES

A. PRIMARY EQUIPMENT

The user manuals noted as references 35 - 43 describe the primary equipment components used for this research, and their general configuration requirements. Figure 6 and reference 27 describe the spectral analysis instrumentation utilized, and schematics for the connection of that equipment for utilization of the resonant dwell technique. This section briefly discusses the resonant dwell apparatus, and provides installation details for the GPIB-PC computer interface board. Set-up parameters for the peripheral equipment controlled by the computer are also covered.

B. SPECTRAL ANALYSIS AND BEAM SPECIMENS

Damping measurements were performed using a modified resonant dwell technique. This method uses forced random vibrations to determine the Specific Damping Capacity (SDC) and Damping Coefficient (zeta = 1/(2 Q)) of cantilever beams by measuring their response to excitation at modal frequencies. (See equations 3 and 7) (Ref. 27: p. 36) The system input and output were measured by accelerometers (Ref. 42) mounted immediately above the beam root and at the beam tip, respectively. The accelerometer outputs were compared (output/input) by a signal analyzer (Ref. 39) to produce the transfer function frequency response for the beam. Based on

this data, Specific Damping Capacity (SDC) and Damping Coefficient (DC) of the beam material was calculated. (Ref. 27: p. 36) Unlike prior research, SDC and DC were calculated by computer at the first modal frequency to demonstrate the functioning of the Signal Analyzer (S/A) Interface Program, the BASIC program developed in conjunction with this report.

The geometry of the cantilever beam specimen is defined in Figure 3. (Ref. 25) Beam width and grip length are specified, but the vibrating length (Lv) and thickness are not. Beams used in this research were originally prepared by LCDR D. Ferguson, USN, (Ref. 8a). Beam dimensions and heat treatments are provided in Table 2 below. All beams were solution treated at the listed temperatures (degrees Centigrade) and furnace cooled.

When using the technique of modal analysis, the fact that beams have multiple resonant frequencies is used to generate significant strains within the beam structure. Forced vibration at one of these resonant frequencies causes certain points (nodes) along the vibrating length to approach their maximum displacement amplitudes. The corresponding shape or response is called the "normal mode" for that resonant frequency. The first three normal modes for a cantilever beam are illustrated in Figure 7. (Refs. 23 and 27) The mode 1 (lowest) resonant frequency is known as the primary (or first) natural frequency, W(n). This first

TABLE 2

Composition (Appendix 1) Specimen #	Thickness (inches)	Width (inches)	Vibrating Length (inches)	Solution Treatment (degrees C)
Fe-Cr-Al				
AB-1	.085	.506	7.087	1100
AB-2	.085	.513	7.071	1100
AB-4	.083	.505	7.071	1050
AB-5	.083	.506	7.087	1050
AB-7	.080	.506	7.063	1000
AB-8	.080	.506	6.909	1000
AB-10	.082	.504	6.929	950
AB-11	.083	.506	6.890	950
Fe-Cr-Mo				
MB-4	.082	.506	7.087	1050
MB-5	.082	.504	7.063	1050
MB-7	.083	.504	7.087	1000
MB-8	.083	.504	7.075	1000
MB-10	.083	.506	7.059	950
MB-11	.084	.504	7.079	950

vibration mode was employed to measure damping in this research.

A photograph of the equipment utilized for this analysis is included as Figure 8. The basic equipment configuration measured the transfer frequency response. The equipment schematic for those measurements is provided in Figure 6.

C. SIGNAL ANALYZER (S/A) & S/A ANALYSIS PROCEDURE

A Scientific Atlanta SD380Z 2-channel signal analyzer (Ref. 39) was used to generate a two volt broadband random noise signal, which was amplified at adjustable gains by a MB Dynamics 2125MB power amplifier (Ref. 41). The amplified signal was wired to drive a MB Dynamics PM-25 Vibramate

Exciter (Ref. 40). This electromagnetic shaker, which is cooled by low pressure air, provides the specimen excitation. The excitation was transmitted to the beam via a rod connected to the base of the beam clamp. The beam clamp assembly provides a 3.5 inch grip length on the beam, with the remainder of the beam free to vibrate. See Figure 3. The clamp jaws are recessed such that the excitation rod, the beam root, and the input sensing accelerometer (mounted atop the clamp) are vertically aligned. The system output accelerometer is mounted at the beam tip. (Ref. 27: p. 37 - 39)

ENDEVCO Model 2250A-10 integral electronics shear accelerometers were used to measure the transfer function (output/input excitation). These generate a voltage that is proportional to their respective acceleration amplitudes. Each accelerometer was attached via a cable to an ENDEVCO Model 4416A signal conditioner. These signal conditioners provide a constant current source of power to the accelerometers, and also amplify the accelerometer output voltage by a factor of ten. The output voltages of the signal conditioners were fed to seperate channels of the signal analyzer. (Ref. 27: p. 42 - 43)

The signal analyzer was programmed to display the material transfer function response on its screen. Specifically, output voltage signals from the accelerometer

located at the beam root were fed into channel A, while signals from the one at the beam tip were led to channel C. The analyzer display depicted non-dimensional transfer function amplitude in db along the vertical axis, with frequency along the horizontal axis.

All displays S/A for first modal responses of the beams listed in Table 2 were recorded on the Zenith computer hard disc, using the Signal Analyzer (S/A) Interface Program developed for this thesis. (See the program explanation later in this report.) To assist with the computer processing of signal analyzer information, all analyzer displays that were compared as a group shared the same set-up page parameters and display screen coordinate dimensions. Further, all were produced with the MB Dynamics amplifier at the same amplification setting.

1. Signal Analyzer (S/A) Set-Up

Eleven different set-up pages are required to program the signal analyzer for operation. Regarding some of the more important parameters listed in those set-up pages, the analyzer voltage signal to the MB exciter was set at 2.0 volts; channel A input level at 0.1V, with channel C at 0.5V; 200 lines of resolution; and averaging set for a 200 target count, using the Hanning weighting function method.

As Cronauer noted (Ref. 27: p. 63), selection of an analyzer signal greater than 2.0 volts to the exciter for a broadband signal type may be prone to uncontrolled amplitude fluctuations, producing erroneous screen displays. Therefore, only 2.0 volt analyzer signals were used for this research.

The 200 target count parameter meant that the analyzer averaged 200 different data samples to establish the displays used for this report.

D. ZENITH COMPUTER

Reference 35 describes the Zenith Z-150 Computer in detail. Briefly, it is an IBM XT compatible machine, employing the Microsoft Corp. MS-DOS Version 3 (series) operating system. (IBM is the registered trademark of International Business Machines Corp.) The computer configuration was modified as the following discussion describes for the purpose of this research.

1. Hard Disc

A Seagate Corp. 30MB hard disc was installed in place of the second Zenith 5 1/4" floppy disc. The hard disc was installed according to the standard procedures given in reference 36. The hard disc was partitioned entirely under the MS-DOS format, and the computer was programmed to load the operating system from the hard disc root directory upon power-up. The S/A Interface Program, the Zenith BASICA

executive program, all analyzer data files, configuration files, etc. were located in a directory entitled "GPIB-PC".

2. Interface Board & Connections

A National Instruments General Purpose Interface Bus (GPIB-PC2A) was installed in a vacant computer utility slot.

a. The GPIB cable was led from the card, out the back of the computer, and to the bus connection on the Scientific Atlanta Signal Analyzer. The analyzer address was reset at its back panel switch station to 25. (Ref. 39, p. 8-1)

b. A second GPIB cable was "piggybacked" from the analyzer bus connection to the Hewlett-Packard Graphics Plotter. This arrangement permitted the plotter to be run by either the computer or the analyzer, though this researcher operated the plotter entirely from computer files. The plotter address was reset using its switch panel to 30. (Ref. 38, p. 9-2 & 3)

E. GPIB-PC CONFIGURATION PROGRAM

GPIB-PC software general installation procedures are covered by pages 2-3 through 2-15 of reference 37. When the operating software for the GPIB-PC is first installed, the configuration routine IBCONF must be run to enter equipment identification addresses and other operating parameters. For this research, the interface board configuration program IBCONF was filed at the root directory on the hard disc. It

is menu driven, and can be activated at the base directory by typing "IBCONF". The GPIB-PC board has the capability to service sixteen programmable instruments, as indicated by the first configuration page of IBCONF. For this report, only the the first two GPIB-PC configuration pages were enacted; DEV 1 for the Signal Analyzer, and DEV 2 for the HP-plotter. The following parameters were entered for this research:

1. GPIB-PC INTERFACE BOARD SET UP PAGE

primary GPIB address	21
timeout setting	T3S
EOS byte	00H
terminate read on EOS	yes
set EOI with EOS on write	no
comparison on EOS	7 bit
set EOI w/last byte of write	yes
GPIB-PC model	PC2A
board system control	yes
local lockout all devices	yes
disable auto serial poll	yes
high speed timing	no
interrupt jumper setting	none
base I/O address	02E1H
DMA channel	none
internal clock	5 MHz

2. GPIB-PC SET-UP PAGE for SIGNAL ANALYZER

{DEV 1 specified as S/A}

primary GPIB address	25
timeout setting	T3S
EOS byte	00H
terminate read on EOS	yes
set EOI w/EOS on write	no
comparison on EOS	7 bit
set EOI w/last byte of write	yes

3. GPIB-PC SET-UP PAGE for HP-PLOTTER

{DEV2 renamed HPPLTR}

primary GPIB address	30
secondary address	none
timeout setting	T3S

EOS byte	00H
terminate read on EOS	no
set EOI w/EOS on write	no
comparison on EOS	7 bit
set EOI w/last byte of write	no

III. SIGNAL ANALYZER INTERFACE PROGRAM

A. GENERAL DESCRIPTION

The signal analyzer interface program written for this thesis is an initial attempt to create a simple program that will: (1) set up the Scientific Atlanta Signal Analyzer; (2) capture and store graphic data compiled by the analyzer; (3) and that will conduct initial analysis of the stored analyzer data. Although this thesis was primarily concerned with computer data acquisition and processing of information associated with vibration damping alloys, most segments of the interface program can be used with any analyzer individual graphic screen display of a two dimensional (X, Y) coordinate function.

The interface program provides the additional capability of deciphering data produced by the signal analyzer into its constituent ASCII characters, and displaying such interpretation on the screen and printer. In conjunction with its data storage feature, the program's ability to interpret the stored data can assist with the composition of other programs to satisfy additional analysis needs.

B. OPERATING SYSTEM AND BASIC

The Signal Analyzer Interface Program is written in GW BASIC by Microsoft Corporation. This BASIC language is functional within the Zenith DOS operating system, supplied

as standard software with Zenith Corporation machines purchased under existing GAO contracts. Importantly, this operating-system and BASIC language are IBM-compatible, which affords additional portability for the interface utility routines. (IBM is a registered trademark of International Business Machines.)

1. <u>BASIC Language</u> The program was written in BASIC for several reasons. Commercial hardware and software were available for use with BASIC. Other languages such as FORTRAN, were also available, but initial evaluation suggested their use would entail additional complexity. The learning curve for this researcher was much shorter, using BASIC, a language with which he was already familiar. Also a consideration, BASIC satisfied the anticipated speed and throughput requirements.

C. INTERFACE BUS (GPIB-PC) and SOFTWARE

The Signal Analyzer interface program relies upon a National Instruments General Purpose Interface Bus designed for IBM Personal Computers and compatibles. This interface bus, or GPIB-PC for short, is a commercially available electronics card that sockets into existing expansion slots in the microcomputer. (GPIB-PC is a registered trademark of National Instruments, Austin, Texas.) This card provides the signal interface necessary for computer-to-Signal Analyzer communications. The GPIB-PC is covered in greater detail in the equipment section of this thesis.

1. GPIB-PC Software

From the programming standpoint, the GPIB-PC is supplied with software that enables direct communications using GW BASIC with programmable instruments like the Scientific Atlanta Signal Analyzer. In effect, the National Instruments GPIB-PC and its accompanying software interpret communications between the two instruments, the computer and the Signal Analyzer. To enable this facility, the GPIB-PC software does require that additional statements and functions be added to the normal BASIC program to execute the interface board communications routines.

For example, the beginning of the BASIC program <MCNFG.BAS>, lines #120 - 220 execute subroutines to operate the GPIB-PC board. Construction of these introductory program lines is provided by the GPIB-PC software. These same lines appear in several of the Signal Analyzer interface sub-programs, and must be included within any BASIC program that communicates with the Signal Analyzer.

As an example of the GPIB-PC functions added to the the BASIC language interpreter, IBWRT(variable, command) is a GPIB-PC function to write instructions to the Signal Analyzer (or other programmable instrument) via the interface board. The "variable" supplies the computer bus address for the analyzer, while the "command" supplies the operator desired instructions to be written to the analyzer.

D. HARD DISC

The system used to construct this program was composed of a Zenith Z-150 microcomputer, purchased under an existing GAO contract, and suitable for shipboard use. It is an IBM-compatible XT-type machine, with a Zenith monochrome (green) high resolution monitor. The system is described in greater detail in the equipment portion of this thesis. It is mentioned here because this system was modified to include a Seagate ST-238 Model 30MB hard disc. The interface program was intended to operate with a hard disc because the data files are large, and the hard disc operates much faster than do floppy discs.

PRIMARY PROGRAMMING FEATURES Subsequent lettered paragraphs under this heading describe some of the primary program features that are common to all the Signal Analyzer Interface Program utility sub-programs executed from the main menu.

E. MENU DRIVEN

The interface program is "Menu Driven," that is, the ten sub-programs that comprise the body of utility functions can be summoned for use from one main menu program. Once the operator has initiated the GW BASIC <TM> program environment, one merely types <RUN "MENU"> to bring the interface program menu on the screen. When activated, the Main Options Menu resides at the hub of the ten utility routines listed by the menu. Selecting a menu option, or choice, calls the selected

sub-program into computer memory and exectutes it. At the conclusion of a sub-program, the menu is redisplayed on the computer screen.

a. This structure was utilized because it was easier to write than one large program that would have attempted integration of all the utility functions together. More importantly, each sub-program was a learning experience for the writer and was used in turn to fashion the next utility sub-program. Thus, the structure evolved sequentially which accounts for its current form.

b. It is stipulated that the interface program could have been restructured, at least in part, to operate faster and more efficiently. However, the interface program is functional and due to its simple structure, the program hopefully will evolve under the scrutiny of subsequent users.

F. LINEAR DESIGN

Each utility routine follows a linear design wherein the raw data is sequentially processed by successive loops and/or sequential minor routines. If the programs are examined (Appendix E), many functional segments can be seen to be delineated by REMARK statements, most of which are labelled as to purpose, such as:

```
REM ******* No Existing File Error Trap *************
```

Frequently, remark statments describe the logic used to construct the functional routines. For example, line #1840 from the sub-program <GRDTA.BAS>:

REM The following three integers institute a "pen down" instruction.

```
B%(11) = 15163 : B%(12) = 17488 : B%(13) = 15163
```

These five digit integers comprise instructions that activate the Hewlett Packard plotter to lower its pen to paper. Though this linear design is uneconomical from a programming standpoint, it will facilitate subsequent users in understanding the program logic and changing it as required for their purposes.

G. INTEGER VARIABLES

Two considerations underlie why integer variables and matrices were frequently employed within the BASIC programs. First, the commercial interface board used in the Zenith Z-150 computer used integer arrays to communicate data between the Signal Analyzer and the computer. Other communications algorithms are also available within the same GPIB <TM> software, but they were more difficult to utilize. Second, integer variables and arrays are more swiftly handled by the computer, which partially mitigates the linear program design.

H. ARRAYS

Throughout the interface program structure, one dimensional integer arrays are used preferentially. The arrays provided convenient "spots" or locations for the collection of data as it accumulated along the linear processing path. It is hoped that these same arrays will make

it possible for subsequent programmers to reprogram for their own needs because each array holds a particular type of data that is described within each segment's code, and by this narrative report. For instance, line #160 of the program <SCRNDTA.BAS> states:

```
REM A%(xxxxx) is a matrix used to store signal analyzer
graphic data from the designated disc file into the computer
active memory
```

Stored on disc directly from the interface board in the computer, this graphic data was produced by the Signal Analyzer, and is in five digit integer format, each integer representing two ASCII characters. When read from disc by <SCRNDTA.BAS>, this graphic data is stored in the matrix A%(xxxxx), each element of which is numbered from one to the end of the file.

Similar usage of matrices is employed through all ten utility programs. If a subsequent user can discern what a matrix contains, he may well be able to modify the existing program to suit his current needs.

I. ERROR TRAPS AND FILENAME EXTENSIONS

Perhaps the most difficult task connected with writing these utility routines was construction of the error trapping sections. Several traps are common to all ten routines; they are summarized as:

1. "No Existing Disc File" is a routine which captures an incorrect operator response to a program request for a disc filename, wherein the filename specified does not exist. The error trap, listed at the end of pertinent programs, provides a response "File not found; try

again." and returns the operator to the original request for a filename. Often, the incorrect response is answered with the expletive "BEEP BEEP!", which no one particularly cares to hear, but it gets one's attention.

2. IF-statements are used to capture less troublesome incorrect operator responses. For example, line #660 in the program <FPLOT.BAS> reads:

```
IF LEN(FILE$) > 12 THEN 580 : REM Limit filename length
to 12 characters.
```

As it implies, this line checks the filename response to ensure that it is 12 characters or less (an IBM DOS convention), and sends program execution back to the original request if the answer provided is greater than twelve characters long. The operator is appropriately "BEEPED" and given another chance.

To limit the required programming, and as this example illustrates, on-screen error messages are not provided for mistakes such as this. The BEEP signals a possible error, and confirmation is provided by reappearance of the incorrectly answered computer question.

3. Filename extensions are used by the program to segregate different types of files. The three types recognized are:

filename.DTA --- signal analyzer integer graphic data

filename.cfg --- signal analyzer integer machine configurations (analyzer set-up instructions)

filename.xyc --- integer files containing graphic X and Y coordinates (that were decoded by the program <DAMPCALC.BAS>)

IF-statements are used here also to test that correct filename extensions have been applied by the operator. If not, the BEEP plus second chance response is enacted by the program.

4. Undoubtedly, there are some operator responses that have not been anticipated that may "BOMB" the program. To guard against those, these instructions attempt to be detailed, and each utility sub-program called from the main menu provides banner instructions regarding that segment's intended use. Further, each computer

requested response provides short details describing the information sought from the operator. If an error does occur that causes program execution to stop and display a BASIC language error message, the routine can be immediately re-run by typing the "F2" key.

J. THE "F9" FUNCTION KEY

All ten utility programs include a program interrupt feature under the "F9" keyboard function key. Pressing the "F9" key during program operation clears the screen, and asks the operator whether he desires to continue, start the current program segment over, or exit to the main menu. If this key is pressed when the computer is requesting information from the operator, the program interrupt will not occur until after the <RETURN> key is pressed. Otherwise, the interrupt occurs immediately, and displays:

"PROGRAM INTERRUPT... "

"Type <RETURN> to resume this program section."

"Type <KK> to start this program section over."

"Type any other key + <RETURN> to exit to main menu."

"?____ "

This feature provides some flexibility for the program operator, and permits changing menu choices even after a program segment has begun execution.

It is worthy of note that, if a disc file is being accessed when the "F9" key is pressed, that file may be left "OPEN" if the operator decides to access the main menu rather than continue execution of the current routine. Typing

"CLOSE" from the command level in BASIC will ensure that all disc files are closed on the currently selected disc.

K. SCREEN CONFIRMATION

Even when an operator input is NOT required, these programs all contain numerous screen messages to advise the operator, so that it is readily apparent the program in use is functioning correctly. For example, when a disc file is loaded by {INTDTA.BAS}, the computer echoes a portion of the loaded disc file to the screen to confirm that the contents are as expected, and that the program is indeed operating. This approach prevents the possibility of the computer "HANGING" or locking up without such a mishap being evident to the operator.

L. RESPONSES & <RETURN>

All keyboard responses requested by the S/A Interface Program require that the <RETURN> key be pressed. A null response (an empty response) is accomplished by only pressing the <RETURN> key.

IV. RESULTS AND CONCLUSIONS

A. DAMPING CAPACITY VS HEAT TREATMENT

In general, the narrow range of heat treatments produced similarly narrow ranges for specific damping capacity and damping coefficient. The Fe-Cr-Al beam samples exhibited a somewhat greater range of values, whereas the Fe-Cr-Mo beam samples were all quite closely grouped. The following table summarizes the Specific Damping Capacity (SDC) and Damping Coefficient (DC) values for the samples analyzed.

TABLE 3

Alloy	Sample Number	Heat Treatment (Degrees Cent.)	SDC (%)	DC
Fe-Cr-Al	AB-1	1100/FC	62.20	4.95E-02
	AB-2	1100/FC	40.33	3.21E-02
	AB-4	1050/FC	52.96	4.21E-02
	AB-5	1050/FC	66.83	5.32E-02
	AB-7	1000/FC	66.34	5.28E-02
	AB-8	1000/FC	50.58	4.03E-02
	AB-10	950/FC	46.87	3.73E-02
	AB-11	950/FC	42.85	3.41E-02
Fe-Cr-Mo	MB-4	1050/FC	62.95	5.01E-02
	MB-5	1050/FC	60.24	4.79E-02
	MB-7	1000/FC	51.87	4.13E-02
	MB-8	1000/FC	63.66	5.07E-02
	MB-10	950/FC	53.55	4.26E-02
	MB-11	950/FC	55.69	4.43E-02

1. Fe-Cr-Al Samples

Solution treatment of the Fe-Cr-Al samples in the 1100 to 950 degrees Centigrade range did not produce great disparity regarding damping capacity over this range. There is some slight optimization noticable in the 1000 -

1050 degree range from results for samples AB-7 and AB-5. SDC and DC noticeably decreased for both samples (AB-10 & AB-11) treated at 950 degrees Centigrade.

2. Fe-Cr-Mo Samples

The heat treatment range from 1100 to 950 degrees Centrigrade produced an equally narrow range of SDC and DC values for the Fe-Cr-Mo beam samples. Similar to the Fe-Cr-Al, the Fe-Cr-Mo samples (MB-10 & MB-11) exhibited a slight reduction in SDC and DC when treated at 950 degrees Centrigrade.

3. Comparison

Values calculated for SDC and DC were comparable for both alloys. The Fe-Cr-Mo appears to produce more consistent values, sample to sample, but that conclusion is probably premature, considering the small number of samples involved.

4. SDC/DC Correction Factors

Values listed in Table 3 above were generated by <GRAPHXYC.BAS>. The SDC/DC correction factors were 2.6523 for the Fe-Cr-Al samples and 2.01912 for the Fe-Cr-Mo. Samples AB-7 and MB-11, respectively, were used for the calculation of these factors. SDC values for these baseline samples calculated directly from the signal analyzer were:

AB-7 .... 66.3442%
MB-11 .... 55.6947%

Calculation of these factors is delineated in detail in Appendix B, specifically the program operating instructions for <DAMPCALC.BAS> and <GRAPHXYC.BAS>.

B. INTERFACE PROGRAM RESULTS Appendices C and D contain all frequency response curves and computer print-outs for the analyzed samples.

1. Appendix C: In addition to individual graphs for each beam, there are two composite graphs combining frequency response curves for samples {AB-1, AB-4, AB-7, AB-10} and {MB-4, MB-7, MB-10} respectively. The individual sample curves were produced by <CPLTR.BAS>. The composite graphs were produced from <CPLTR.BAS> and <GRDTA.BAS>.

2. Appendix D: SDC and DC computer calculated values included as program readouts in Appendix D were generated by <GRAPHXYC.BAS>. That sub-program was used vice <DAMPCALC.BAS> because it contains a curve smoothing routine that reduces inaccuracies due to "jagged" or noisy signal analyzer curves.

C. IMAGING FERROMAGNETIC DOMAINS

When this report was prepared, it was not possible to image the ferromagnetic domain walls using the transmission electron microscope at the U. S. Naval Postgraduate School. The lower lens magnetic field extended through the specimen and oriented the domain walls such that the electron beam was not sufficiently deflected to produce an image.

APPENDIX A

METALLURGICAL ANALYSIS REPORTS

LABORATORY CERTIFICATE

Anamet Laboratories, Inc.

3400 INVESTMENT BOULEVARD • HAYWARD, CALIFORNIA 94545-3811 • (415) 887-8811

APPENDIX A

Laboratory Number: 587.071
Purchase Order: N62271-87-M-2087
Requisition No: N62271-7117-5066
Date Submitted: May 8, 1987
Date Reported: May 18, 1987

Naval Postgraduate School
Supply Officer - N62271
Attn: Dr. J. Perkins/Georgia Gooder
Receiving Officer Bldg. 349
Monterey, CA 93943

------- CAPTION -------

Sample #287 was the Fe-Cr-Mo alloy. The composition was unknown by the laboratory at the time when analysis was conducted on both samples. Sample #460 was the Fe-Cr-Al alloy.

SUBJECT

Two metal coupons were submitted for chemical analysis. The samples were identified as follows: 287 and 460.

CHEMICAL ANALYSIS
(reported in wt. %)

Mark:		Fe-Cr-Mo 287	Fe-Cr-Al 460
Aluminum	(Al)	0.01	2.99
Carbon	(C)	0.009	0.002
Chromium	(Cr)	11.87	11.82
Copper	(Cu)	<0.01	<0.01
Manganese	(Mn)	<0.01	<0.01
Molybdenum	(Mo)	2.93	0.01
Nickel	(Ni)	<0.01	<0.01
Phosphorus	(P)	0.007	0.007
Silicon	(Si)	<0.01	<0.01
Sulfur	(S)	0.005	0.005
Titanium	(Ti)	<0.01	<0.01
Vanadium	(V)	<0.01	<0.01

This testing was performed in accordance with the purchase order.

Submitted by;

E. A. Foreman
E. A. Foreman
Manager, Quality Control

3c/bh51587

Luvak Inc.

722 Main Street
Boylston, MA 01505
Tel. (617) 869-6401

Analytical Report No. 0-1500

REQUESTED BY David W. Taylor, Naval Ship R&D Center
Annapolis Laboratory
Annapolis, Maryland 21402

ATTENTION C. Wong
Code 2812
Bldg. 47, Rm. 3F

Invoice Date 10/31/86

Invoice Number	Customer's Order Number	Customer's Requisition Number	Date Received
05907	N61533-84-A-0087-V717		10/23/86
		Job order #1-2803-107-06	

DESCRIPTION

4 Samples, Stainless Steel, analyzed for elements listed below.

RESULTS

------- CAPTION -------

Fe-Cr-Al sample analysis for sample taken from the edge of the ingot.

SAMPLE IDENTIFICATION

2E edge

	%		%
Carbon	.007	Nickel	.006
Nitrogen	.0011	Copper	<.001
Oxygen	.0019	Cobalt	<.001
Aluminum	2.89	Vanadium	<.001
Molybdenum	<.001	Titanium	<.001
Sulfur	.004	Iron	84.9
Chromium	11.61	Hydrogen	.0001
Manganese	<.001	Platinum	<.002
Silicon	<.002	Boron	.002
Phosphorus	<.002	Calcium	.0018

LUVAK INC.

Luvak Inc.

722 Main Street
Boylston, MA 01505
Tel. (617) 869-6401

Analytical Report No. 0-1500

Carl C. Wong

REQUESTED BY David W. Taylor, Naval Ship R&D Center
Annapolis Laboratory
Annapolis, Maryland 21402

ATTENTION C. Wong
Code 2812
Bldg. 47, Rm. 3F

Invoice Date 10/31/86

Invoice Number	Customer's Order Number	Customer's Requisition Number	Date Received
05907	N61533-84-A-0087-V717		10/23/86
		Job order #1-2803-107-06	

DESCRIPTION

4 Samples, Stainless Steel, analyzed for elements listed below.

RESULTS

SAMPLE IDENTIFICATION

3C Center

CAPTION
Fe-Cr-Mo sample analysis for sample taken from the center of the ingot.

	%		%
Carbon	.006	Nickel	.005
Nitrogen	.0012	Copper	<.001
Oxygen	.021	Cobalt	<.001
Aluminum	.002	Vanadium	<.001
Molybdenum	2.92	Titanium	<.001
Sulfur	.003	Iron	84.8
Chromium	11.65	Hydrogen	.0001
Manganese	<.001	Platinum	<.002
Silicon	.010	Boron	.002
Phosphorus	<.002	Calcium	.0014

LUVAK INC.

Luvak inc.

722 Main Street
Boylston, MA 01505
Tel. (617) 869-6401

Analytical Report No. 0-1500

REQUESTED BY David W. Taylor, Naval Ship R&D Center
Annapolis Laboratory
Annapolis, Maryland 21402

ATTENTION C. Wong
Code 2812
Bldg. 47, Rm. 3F

Invoice Date 10/31/86

Invoice Number	Customer's Order Number	Customer's Requisition Number	Date Received
05907	N61533-84-A-0087-V717		10/23/86
		Job order #1-2803-107-06	

DESCRIPTION

4 Samples, Stainless Steel, analyzed for elements listed below.

RESULTS

--------CAPTION--------
Fe-Cr-Al sample analysis for sample taken from the center of the ingot.

SAMPLE IDENTIFICATION

2C center

	%		%
Carbon	.007	Nickel	.010
Nitrogen	.0010	Copper	<.001
Oxygen	.0014	Cobalt	<.001
Aluminum	2.91	Vanadium	<.001
Molybdenum	<.001	Titanium	<.001
Sulfur	.005	Iron	85.0
Chromium	11.44	Hydrogen	.0001
Manganese	<.001	Platinum	<.002
Silicon	<.002	Boron	.002
Phosphorus	<.002	Calcium	.0016

LUVAK INC.

By

Luvak Inc.

722 Main Street
Boylston, MA 01505
Tel. (617) 869-6401

Analytical Report No. 0-1500

Page 4 of 4

QUESTED BY David W. Taylor, Naval Ship R&D Center
Annapolis Laboratory
Annapolis, Maryland 21402

ATTENTION C. Wong
Code 2812
Bldg. 47, Rm. 3F

Invoice Date 10/31/86

Invoice Number	Customer's Order Number	Customer's Requisition Number	Date Received
05907	N61533-84-A-0087-V717		10/23/86
		Job order #1-2803-107-06	

DESCRIPTION

4 Samples, Stainless Steel, analyzed for elements listed below.

RESULTS

---- CAPTION-------
Sample analysis for Fe-Cr-Mo sample taken from the edge of the ingot.

SAMPLE IDENTIFICATION

3E edge

	%		%
Carbon	.006	Nickel	.006
Nitrogen	.0009	Copper	<.001
Oxygen	.020	Cobalt	<.001
Aluminum	.002	Vanadium	<.001
Molybdenum	2.92	Titanium	<.001
Sulfur	.005	Iron	84.7
Chromium	11.70	Hydrogen	.0002
Manganese	<.001	Platinum	<.002
Silicon	.011	Boron	.002
Phosphorus	<.002	Calcium	.0016

LUVAK INC.

APPENDIX B

SIGNAL ANALYZER PROGRAM DESCRIPTION
UTILITY SEGMENTS 1 - 10

A. MAIN MENU The Main Menu program is entitled {MENU.BAS}, and is located in the hard disc directory <GPIB-PC>. It is activated from the GW BASIC environment by typing "RUN MENU". In response to that command, the following banner menu appears:

```
***************************************************************
**            SIGNAL ANALYZER INTERFACE PROGRAM              **
**                   MAIN OPTIONS MENU                       **
**                                                           **
**     <MCNFG.BAS> ........................  1               **
**     <FPLOT.BAS> ........................  2               **
**     <PRPLOT.BAS> .......................  3               **
**     <CPLTR.BAS> ........................  4               **
**     <INTDTA.BAS> .......................  5               **
**     <GRDTA.BAS> ........................  6               **
**     <SCRNDTA.BAS> ......................  7               **
**     <ITERPLOT.BAS> .....................  8               **
**     <DAMPCALC.BAS> .....................  9               **
**     <GRAPHXYC.BAS> ..................... 10               **
**     EXIT THIS MENU ..................ANY OTHER KEY        **
**                                                           **
**                  CHOICE? ___                              **
**                                                           **
**                                                           **
**     PERKINS/PATCH MASTERS THESIS {NAVPGSCOL}              **
**     COPYRIGHT AUGUST 20, 1987 UNITED STATES NAVY          **
***************************************************************
```

The blinking cursor by "CHOICE" awaits the operator's election, an integer from one to ten, or "ANY OTHER KEY" plus RETURN>. These displayed options are the only possible esponses recognized by the <MENU.BAS> program.

Each numbered choice plus <RETURN> activates a separate sub-rogram, all of which are explained in subsequent paragraphs. yping any other key plus <RETURN> exits the Signal Analyzer nterface Program, but does not remove the menu from the computer emory. Thus, typing "RUN" or pressing the F2 key will redisplay he main menu.

. MACHINE CONFIGURATION {MCNFG.BAS} The machine configuration rogram is located under option #1 on the main menu. This BASIC rogram sets up the Signal Analyzer (S/A), or it stores the urrent Signal Analyzer configuration (set-up) on hard disc.

1. Selecting option #1 from the main menu and pressing

<RETURN> displays the following banner instruction.

```
********* Machine Configuration Program <MCNFG.BAS> ***********
** This program can store signal analyzer machine configuration **
** codes obtained originally from the analyzer itself.  The     **
** program can also set up the signal analyzer, using any       **
** configuration data files previously saved to computer disc.  **
******************************************************************
**   Ensure that the Signal Analyzer is connected and  ON       **
**   and that its GPIB address is 25.                           **
******************************************************************

Type  1  to obtain & store the current configuration.
Type  2  to set up analyzer with an existing config'tn file.

1 or 2 ... ? ___
```

2. Signal Analyzer (S/A) requirements to use the Interface Program machine configuration routine are relatively simple.

a. The Scientific American Signal Analyzer (Model SD 380Z) operating manual provides instructions to set the machine interface buss address to 25. Selecting setup page #8 from the Signal Analyzer front control panel produces the IEEE COMMUNICATIONS setup page, the first entry of which is the Signal Analyzer address option. Note that changing this address to 25 disables the Signal Analyzer's ability to operate the Hewlett Packard graphic plotter. However, the Signal Analyzer Interface Program can operate the plotter instead.

(1) The Signal Analyzer operating manual also describes how to set switches in the back of the machine which fix its address to the chosen value each time the machine is turned on. This is a more convenient method, and the recommended method for use with the Interface Program. Setup page #8 preserves the operator's option to reset the Signal Analyzer address as required after power is applied.

3. With the Signal Analyzer turned on and its address correctly set, {MCNFG.BAS} option 1 will store the current Signal Analyzer machine configuration (all eleven set-up pages, plus screen format) to hard disc.

a. In response to an option 1 selection, the program obtains the Signal Analyzer configuration data in integer format, displays a sample of the data to the screen, confirming successful communications, and then requests...

```
"What file name for disc storage?"
"Note: The file designation must be  cfg  (ie); filename.cfg"
"Type  <RETURN>  to bypass disc storage?"

"Filename.cfg ... ?" ___
```

Filenames may be anything the operator desires, except

that the file designation must be "CFG", and filename lengths must be twelve characters or less, including the period and designation. The "CFG" extension is used by the Interface Program to keep files segregated regarding purpose, and to prevent erroneous selection of configuration files by other program segments.

Typing <RETURN> only bypasses the file storage function, and brings the following prompt to the screen:

"Type yes to run program again. " ?___

Any response beginning in "Y" (and merely "Y" itself) will redisplay the {MCNFG.BAS} banner and provide another opportunity to execute this Interface Program segment.

4. Selecting option 2 will display the Signal Analyzer configuration files previously stored to hard disc. The operator will be asked:

"What file name for retrieval from disc?"
"Note: The file designation must be cfg (ie); filename.cfg " ?___

Input of a correct filename causes the Interface Program to obtain the disc file, display a sample of its contents, then request:

"Type yes to set up signal analyzer using this file. " ?___

Again, any response beginning with "Y" will send the selected configuration data to the Signal Analyzer. Any other response will bypass the set-up function.

Following transmission of the configuration data to the Signal Analyzer, the prompt appears:

"Type KK + <RETURN> to reset using same file. "?___

This prompt permits retransmission of the same set-up parameters to the Signal Analyzer if, for some reason, the prior transmission was not received. Any response different than "KK" will bypass this option.

a. Note that if the Signal Analyzer is currently displaying a set-up page menu, it will not be apparent from the Analyzer screen that any configuration instructions were received from the computer. However, changing the Analyzer screen display to graphics will show the computer transmitted graphic set-up format, and changing back to the Analyzer set-up page will display the transmitted set-up parameters, confirming that the computer file was transmitted.

5. A response not equal to "1" or "2" will redisplay the {MCNFG.BAS} banner. However, pressing the "F9" key and <RETURN> will display mid-program exit instructions as discussed in the prior GENERAL DESCRIPTION.

6. Following completion of option "1" or "2", the operator is asked:

"Type yes to run program again. "?___

Any response beginning with "Y" will rerun the {MCNFG.BAS} sub-program. Any other response (except for F9) will exit this section and display the main menu program {MENU.BAS}.

C. FILE PLOT {FPLOT.BAS} This sub-program files Signal Analyzer (S/A) screen images as integer hard disc files that are named by the operator. Any Signal Analyzer graphic screen plot can be filed.

1. Selection of main menu option 2 produces the following computer screen display:

```
*****************  PROGRAM  <FPLOT.BAS>  ***********************
** This program files signal analyzer screen images in disc    **
** files named by the operator.  Each analyzer screen image is **
** stored as an integer array.   Each element of the integer   **
** array represents two ASCII code characters from the HP      **
** plotter language. (Ref: HP Plotter Prog. Manual, pg. 1-8 & 9)**
** Please ensure that the signal analyzer is turned on and     **
** that it's GPIB address is twenty-five (25).                 **
*****************************************************************
** NOTE: Use <MCNFG.BAS> to store Signal Analyzer machine      **
** configuration files identified as  ''filename.cfg''         **
*****************************************************************
```

"What filename do you wish to specify?"
"Note: File designations must be dta (i.e.); filename.dta "

"Type your filename.dta " ?____

A filename response with a designation of "DTA" will cause the routine to obtain the Signal Analyzer (S/A) screen display information, display sample integer data to the computer screen, then display confirmation that the disc filename specified contains a certain number of elements and has been closed.

2. Relevant considerations are:

a. Dual Signal Analyzer screen displays may be filed, but they will not be correctly processed by the S/A Interface sub-programs {GRDTA.BAS} or {DAMPCALC.BAS}.

b. Regardless of what image or menu/set-up page is displayed on the S/A screen, this program obtains whatever is in the graphic display memory of the analyzer.

c. Each disc file of a S/A graphic screen contains about 2400 to 4600 five digit integer elements. {FPLOT.BAS} does

.imit the file lengths to the minimum possible, using the integer :ormat.

3. The Machine Configuration sub-program {MCNFG.BAS} should)e used to file S/A configuration data. (See paragraph B)ove.)

). PRINTER PROGRAM {PRPLOT.BAS} Selection of option #3 from the ıain menu executes the {PRPLOT.BAS} sub-program. This routine sends :he contents of a GPIB-PC integer disc file to the printer, while lisplaying the same contents to the computer screen. The sub-program /ill print the contents of any integer disc file containing less than .0,000 elements. However, it was specifically designed to print the :ollowing file types:

* filename.DTA ... Integer data files holding Signal Analyzer graphic screen displays

* filename.CFG ... Integer data files holding Signal Analyzer machine configurations

* filename.xyc ... Integer data files holding the X & Y coordinates (in HP-plotter terms) of Signal Analyzer graphic screen displays

1. Main menu option #3 causes the following instructions banner :o be displayed:

```
**************** Program Name <PRPLOT.BAS> ***********************
** This program prints the contents of a GPIB disc file on the  **
** line printer.  The integer disc file is printed in the same  **
** format in which it was stored.  Any integer disc file        **
** containing less than 10000 integer elements can be printed.  **
******************************************************************

"Type <RETURN> to continue... " ___
```

Typing <RETURN> clears the screen, and displays the current :xisting disc files of the type described above. Further, the ›perator is asked:

```
"What filename?"
"Please include a file designation (i.e.); filename.XXX "
  "where  XXX  is the designation."
"[----NON-INTEGER files will not correctly load. ----]

"Filename.xxx " ?___
```

a. An incorrect filetype response {such as BASIC filename.BAS), or EXECUTIVE (filename.EXE)} will cause the nstructions banner to be redisplayed, providing the operator another hance to correctly respond.

b. A correct filetype response will load the file from hard disc and send its contents to the printer and computer screen.

2. If the operator does not desire the entire file contents, typing the "F9" function key will interrupt the routine, and display the mid-program exit options. The printer, however, may contain data in its printer buffer, so several more pages may be typed to paper unless the printer is turned off.

a. Note; turning the printer off while {PRPLOT.BAS} is operating MAY cause the program to "HANG"; that is, the computer may lock up. Turning the printer back on usually clears this condition. This condition does not occur when the "F9" function key exit option is enacted.

b. If the printer is never turned on and {PRPLOT.BAS} is executed, the program only displays file contents to the computer screen.

3. After printing the contents of a disc file, {PRPLOT.BAS} will display the prompt:

"Type yes to print another file " ?___

Any operator response beginning with "Y" will redisplay the beginning instructions banner for {PRPLOT.BAS}. Any other answer will send the operator to the main menu.

E. COMPUTER PLOTTER {CPLTR.BAS} This S/A Interface sub-program loads the contents of hard disc graphic files created by {FPLOT.BAS} and produces a hard-copy graph, using the Hewlett Packard plotter.

1. Selection of main menu option 4 displays the banner instructions:

```
************* Program Computer Plotter <CPLTR.BAS>  ***************
** This program uses the HP-plotter to draw a graph using data   **
** from a computer disc file selected by the operator.  The file **
** chosen must contain an image from the Scientific Atlanta      **
** signal analyzer that was saved to disc in integer format.     **
** The file designation must be {.dta}, (i.e.) filename.dta      **
** Please ensure that the plotter is connected and turned on.    **
*******************************************************************
```

"Type <RETURN> to continue... "

Pressing the <RETURN> key clears the screen and displays:

"What file name contains your graph" ?___

Upon entry of a valid filename, the {CPLTR.BAS} routine responds:

"Loading disc file ";FILE$;" ... Please wait... "
"++++++++++++++++++++ ...

"++++++++++++++++++++ ...

When the file is loaded into the computer, the program advises:

"Plotting the disc file (filename.DTA)"

2. Upon completion of the hard-copy graph, {CPLTR.BAS} advises the number of instruction bytes written (transmitted) to the plotter, and the plotter status code, according to Table 4.1 of the National Instruments GPIB-PC Operating Manual.

3. At the conclusion of the aforementioned operations, {CPLTR.BAS} exhibits the prompt:

"Type yes to run this program again."
"Type K to print the same disc file again." ?___

a. The "K" option makes it possible to print numerous copies of the same graph before restoring the computer memory erases the graphic instructions.

b. Any input beginning with "Y" starts {CPLTR.BAS} over.

c. Any other response exits to the main menu.

4. The only preparations required for the Hewlett Packard 7470A Plotter are: (1) load a pen into the left pen recepticle, (2) position the paper against the loading stop, (3) and turn on the plotter.

a. Of course, these preparations must be repeated for each new graphic image plotted. And, it is assumed that the GPIB-PC IEEE interface configuration program has been configured as described in the Equipment Section of this report.

F. INTERPRET DATA {INTDTA.BAS} This S/A Interface sub-program converts the integer contents of a GPIB-PC disc file into ASCII characters for operator interpretation. The converted code is displayed to the screen as well as printed out by the line printer.

1. Signal Analyzer (S/A) graphic screen displays, when decoded, are written in Hewlett Packard graphic plotter language. The HP-7470A Graphics Plotter Interfacing and Programming Manual contains the syntax of the HP plotter language.

2. Signal Analyzer machine configuration files, when decoded, are written in code unique to the Scientific Atlanta Signal Analyzer, Model SD380Z. The Scientific Atlanta S/A Operating Manual, Appendix . contains the syntax for this code.

3. Selecting option 5 on the main menu will execute the {INTDTA.BAS} sub-program, which will display the following instructions banner:

```
******************Program Name <INTDTA.BAS> ********************
** This program converts  integer  contents of a GPIB disc file **
** to ASCII characters for interpretation.  The converted code  **
** is displayed on the screen and output to the line printer.   **
** (Non-integer files won't load.)  {Interpretation of output:  **
** Plotter language  is  explained  in HP Plotter Prog. Manual  **
** & Signal Analyzer codes  in  Sci. Atlanta Operating Manual.} **
*****************************************************************
** NOTE: Type F9 to interrupt program printing operation if the **
** entire decoded file is not desired from the line printer.    **
*****************************************************************
```

"Type <RETURN> to continue... " ____

a. Pressing the <RETURN> key clears the screen and displays existing files in the disc directory, followed by the prompt:

```
"What filename? "
"Please include a file designation  (i.e.); filename.xxx  "
" where  xxx  is the designation (BAS, BAT, COM or
                                       EXE are invalid)."
```

"Filename.xxx" ?____

A correct filename will cause the loading of that file, and display of a portion of its integer contents. Then, {INTDTA.BAS} will interpret the file contents and output its results to the computer screen, as well as to the printer.

b. {INTDTA.BAS} output has the following format: (example)

```
53216 43878 21342 56564 ; ; P A 3 4 0 0
[--- GPIB INTEGERS ---] [= ASCII CODE =]
```

Four integers are followed by their respective interpretations in ASCII code, listed in the same order as the parent integers.

(1) The integer coding algorithm for this sub-program and all others in the S/A Interface Program as well is:

```
          Integer  Integer   ASCII Characters
           17488    15163         P D   ; ;

where ...  BB = INT(17488/256) and  AA = 17488 - (BB * 256)
           P = CHR$(AA)  and   D = CHR$(BB)

           DD = INT(15163/256) and  CC = 15163 - (DD * 256)
           ; = CHR$(CC)  and  ; = CHR$(DD)
```

c. Integer file types other than filename.DTA or filename.CFG may be loaded by the program, but their interpretation

ill probably not yield any useful information, unless they were roduced by other Signal Analyzer resident software.

4. An incorrect or non-existent filename will trigger the edisplay of the {INTDTA.BAS} instructions banner, providing the perator another chance to input correct information.

5. Operation of this sub-program is similar to {PRPLOT.BAS} in hat the operator may use the "F9" program interrupt key to abort its utput if the entire interpreted file is not desired. As for that rogram, turning the printer OFF during output operations may make he computer system "HANG" , that is, lock up. Turning the printer ack ON usually clears this condition. If the "F9" interrupt is sed, this malfunction should not occur.

6. After selected disc files are interpreted and printed, INTDTA.BAS} displays the prompt:

"Type yes to print another file" ?____

s for all the S/A Interface sub-programs, this prompt will accept ny response beginning in "Y" as an affirmative answer, and will eturn execution to the {INTDTA.BAS} instructions banner. At that oint, the operator has an additional opportunity to interpret nother disc file. Any other response will exit this sub-program and eturn the main menu to the screen.

. GRAPH DATA {GRDTA.BAS} Comparison of amplitude verses frequency esponse for several samples can be conducted using this sub-program. GRDTA.BAS} loads the contents of a GPIB-PC disc file; identifies the ortion of that file containing just the graphic curve; then plots he curve only, using the Hewlett Packard plotter. Thus, if CPLTR.BAS} is first used to draw the entire grid background and the irst curve, {GRDTA.BAS} can add additional curves (without their espective coordinate grids) for direct comparison. Of course, all urves displayed together must have originated from S/A screen isplays with common coordinate grids.

1. Main menu option 6 will call the following {GRDTA.BAS} nstructions banner to the computer screen:

```
**************** Program <GRDTA.BAS> ******************************
* This program loads the contents of a designated GPIB disc file **
* containing integer data into the computer; identifies the file **
* graphic data section, then plots that disc file graphic data   **
* using the Hewlett Packard plotter.  This program can be used   **
* to plot several signal analyzer graphic displays on the same   **
* page, providing direct comparison of different sample results. **
*******************************************************************
* NOTE: Curves plotted together on the same plotter display      **
* should all have originated from equivalent coordinate scales   **
* when they initially were displayed on the Signal Analyzer.     **
*******************************************************************
```

"Type <RETURN> to continue... " ?___

Pressing the <RETURN> key will clear the screen, display the current graphic data files, and request:

"What file name contains your data?"
"File designation must be dta (ie); filename.dta "
"Note: File desig's <EXE>, <BAS>, <BAT> & <COM> will not load."

"Filename.dta ..." ?___

2. Entering a valid filename will start the {GRDTA.BAS} sub-program execution. As the program operates, it will exhibit screen messages describing its operational stages. At the point where the selected graph is plotted, the routine displays the screen message:

"Plotting the selected graphic data ... "

"Calling IBWRTI(B%(matrix))... "
"The interface board function to plot the data... "

This message refers to the GPIB-PC computer interface board and its software function IBWRTI( ) that outputs the graphic data (stored in B%(matrix)) to the HP-plotter.

3. At the conclusion of the plotting operation, the routine {GRDTA.BAS} displays the location within the selected file of the graphic information. The index numbers of the integers containing the start and end of the graphic curve are shown on the computer screen by the prompt:

"Starting integer # = " (start index) "Ending integer # = " (ending index)

The numbers provided begin with the first integer of the disc file.

4. The following prompt is also displayed at this time:

"Type <KK> to RE-PLOT the same graph."
"(Reposition graph paper to origin for a re-plot.)"

"Type yes to run the program again..." ?___

a. As it indicates, a response of <KK> will replot the same graphic information. Reposition the paper to its top edge and the pen to the left side of the plotter (HP Plotter Op. Manual, position P1) before this operation occurs or an unwanted line will result when the pen is drawn backwards across the page.

b. An entry of <YES> or any entry starting with "Y" will restart {GRDTA.BAS} by clearing the screen and displaying the starting instructions banner.

c. Any other entry will exit this sub-program and return he operator to the main menu.

. SCREEN DATA {SCRNDTA.BAS} This program interprets and displays he contents of an integer GPIB-PC disc file on the computer screen. t provides the additional facility of being able to select at will he portions desired for interpretation and display. The operator an move about within the data file contents as necessary to examine hatever portion is required. Since the entire disc file is loaded nto computer RAM, the routine is fairly rapid once the disc file has een loaded.

1. Selecting main menu option 7 activates the {SCRNDTA.BAS} ub-program. The following instructions banner is shown on the omputer screen:

```
************ Program to display disc file <SCRNDTA.BAS> ************
** This program loads the contents of a designated GPIB disc file **
** containing integer data into the computer and then displays    **
** the contents and its decoded ASCII characters for analysis.    **
** Note: File desig's can not be <BAT>, <BAS>, <EXE> or <COM>.    **
********************************************************************
```

"Type <RETURN> to continue... " ?____

1. Typing <RETURN> clears the screen and exhibits the prompt:

"What file name contains your data" ?____

correct filename entry will cause the designated file to be loaded nto computer memory. An incorrect or non-existent filename will rigger redisplay of the instructions banner.

2. When the disc file is successfully input to the computer, he prompt occurs:

"Disc file " <filename> "is loaded into active memory ..."

"Designate the number of the starting byte... " ?___

indful that integer disc files containing graphic data are pproximately 3600 - 8200 integer elements long, enter the integer ndex number of the desired starting point. The program will request ne number again if an incorrect value outside the file boundaries is ttempted. If a character value is entered, the BASIC operating ystem requests re-entry.

3. After selection of the starting integer value, the next rompt is:

"Designate the number of the ending byte... " ?___

he same type of conditions apply here as for the starting index

number. {SCRNDTA.BAS} will request the entry again if the stipulated integer number is outside the file boundaries, and also if the number chosen is negative or less than the chosen starting index.

4. With the starting and ending indexes chosen, {SCRNDTA.BAS} locates the designated information, interprets it, and prints to the screen:

"Your chosen data bytes contain... "

"Byte #	Byte #	Integer	Integer	ASCII Characters "
1	2	17235	11313	S C 1 ,
3	4	12345	12336	9 0 0 0
5	6	12588	12844	, 1 , 2
7	8	13621	17467	5 5 ; D
9 ... (etc.)				

The numbers displayed here are examples of how the interpreted results appear. The computer screen fills with nineteen lines of information, then requests that <RETURN> be pressed to continue with the next screen. This method permits the operator to step through the displayed information at a readable pace.

a. If the "F9" interrupt key is pressed, the operator may choose to continue with the current file; examine another disc file; or exit to the main menu.

5. When all chosen information has been interpreted to the screen, the following prompt appears:

"Type yes to look at other data bytes... " ?___

At this point, other sections of the file held in computer memory may be examined. This feature permits the operator the flexibility to move about within the chosen file.

6. Entry of <RETURN> or any key plus <RETURN> activates the prompt:

"Graphic file contained " <number> " total bytes of data."

"Type yes to examine another disc file... " ?___

An affirmative reply will re-exhibit the {SCRNDTA.BAS} instructions banner, at which time the operator may press <RETURN> and select another file for examination. Any other response not starting with "Y" will return execution to the main menu.

7. During operation of the {SCRNDTA.BAS} routine, when a hard copy is desired of information shown on the screen, the <PrtSc> key

an be used to send it to the printer. Also, <Ctrl><PrtSc> will send ll screened information to the printer, if desired. Typing Ctrl><PrtSc> again releases this selection.

. ITERATIVE PLOT {ITERPLOT.BAS} With this S/A Interface sub-rogram, any chosen portion of an integer graphic file may be drawn ith the HP-plotter. {ITERPLOT.BAS} loads the disc file, requests dentification of desired sections from the operator, then plots the hosen information. During development of the Signal Analyzer nterface Program, this routine was used to gain necessary insight nto the construction of graphic data from the analyzer.

1. Operation of {ITERPLOT.BAS} is very similar to SCRNDTA.BAS}. Information requested from the operator, and the rogram screen prompts requesting that information are similar. herefore, explanations listed in this section are appropriately bbreviated where clarity is not compromised.

2. Choosing entry 8 from the main menu brings the following nstructions banner to the screen:

```
***************** Program <ITERPLOT.BAS> ************************
** Program to selectively plot data from an operator desig'd  **
** disc file.  This program runs the HP-plotter to draw a     **
** graph using designated portions of graphic data from a     **
** disc file that was originally obtained from the Scientific **
** Atlanta signal analyzer screen display by <FPLOT.BAS>.     **
****************************************************************

        "Type <RETURN> to continue... " ?___
```

ne <RETURN> key causes display of the request:

```
    "What disc file do you wish to plot? "
    "Note: File designation must be  dta  (i.e.) filename.dta "

    "What filename.dta   " ?___
```

3. Subsequent to selection of the disc file and its input to the omputer, the following message appears:

```
 "Note: The first several data bytes from the disc file contain
  graphic scale and pen positioning instructions.  Thus, you need
  at least the first ten bytes to produce a readable graphic
  segment on the HP plotter."

 "Designate the number of bytes you wish to use from the
  beginning of the disc file {scale, border, etc.} "

    ? ___
```

s the above instructions imply, the first eight to twelve elements of ne integer graphic data file contain scaling and plotter pen posi-ioning instructions. This screen input requests the operator to

designate how many of the initial file elements he wishes to use. A zero or null input is accepted; however, the subsequent plotter image may not be discernable without the omitted scaling instructions.

a. Negative values and those greater than the original file length will be rejected, and the above prompt redisplayed.

4. The next screen message is:

"Designate the starting byte for the graphic data... " ?___

Followed by:

"Designate the ending byte for the graphic data... " ?___

Upon completion of these messages, the computer program advises:

"Plotting the rearranged data... "

At this time, the selected file sections are drawn by the HP-plotter. Since key elements of the plotter instructions may have been omitted by the operator's choices, the plotter image may be confused. The utility, however, of this program is that it provides graphic illustration of the composition of graphic files produced by the Signal Analyzer.

5. When the plotter completes its operations, the computer screen states:

"plotter status... " (number)

"Type K + <RETURN> to plot the same graph. "

"Type I + <RETURN> to select and plot different portions" of the same graph. "

"Type <YES> + <RETURN> to select another disc file."

"Type <RETURN> to exit to main menu. " ?___

a. Plotter status is a GPIB-PC interface board code, listed in Table 4.1 of the National Instruments GPIB-PC Operating Manual. It is useful as a programming tool.

b. The remaining screen messages provide the opportunities to re-plot the same information; select different portions of the same file, now resident in the computer memory; select another disc file for {ITERPLOT.BAS} to process; or to exit to the main menu.

J. DAMPING CALCULATION {DAMPCALC.BAS} This sub-program is the longest and longest running of the Signal Analyzer (S/A) Interface Program routines. It combines many of the features from previously discussed sub-programs. It also deciphers the X and Y coordinates from within

ny Signal Analyzer single graphic image file. From those oordinates, {DAMPCALC.BAS} also calculates the Specific Damping Capa- ity (SDC) and Damping Coefficient (DC).

1. The computer calculated values for SDC and DC are relevant 'or nodal frequency curves of amplitude verses frequency.

2. <DAMPCALC.BAS.> will correctly decipher and store the integer : and Y coordinates from any single S/A curve. However, it was de- igned for amplitude verses frequency curves with a single maximum, so ts SDC and DC calculations will not produce meaningful results for ther curve shapes. The sub-program should, nevertheless, operate ithout program fault on other curve shapes.

3. Since the analyzer produces somewhat "jagged" curve shapes, epending on its configuration and the material tested, SDC and DC esults form <DAMPCALC.BAS> may be influenced by such irregularities. GRAPHXYC.BAS> employs a curve smoothing routine to reduce the effect f such jagged curves.

4. The deciperhered X and Y coordinates are in terms of the ewlett-Packard graphic plotter scale. With sufficient knowledge of he original graph, these coordinates can be interpreted to other oordinate scales. In any event, the coordinates produced by DAMPCALC.BAS} accurately represent the original Signal Analyzer dis- lay, the curve portion of which can be reproduced on the computer by GRAPHXYC.BAS} (Main Menu selection 10).

5. Main menu selection 9 calls the following banner to the omputer screen:

```
**************** Program <DAMPCALC.BAS> ***************************
* This program loads the integer contents of a designated GPIB    **
* disc file into the computer; identifies the file graphic data   **
* section; then calculates the Specific Damping Capacity (SDC)    **
* and the Damping Coefficient (DC) for the selected data file.    **
* SDC & DC are calculated in absolute HP-plotter coord terms      **
* and may vary some from Signal Analyzer values. This program     **
* will also store the S/A graphic XY-coordinates as absolute      **
* HP-plotter integer magnitudes, if desired.  {The program        **
* <GRAPHXYC.BAS> can display stored disc file graphic data on     **
* the microcomputer screen. (Selection #10 from the Main Menu)}   **
*******************************************************************
* Note: This program will not function correctly with signal      **
* analyzer dual screen traces that are stored to disc.  GPIB      **
* disc data files must be single traces of amplitude vs freq.     **
*******************************************************************
```

"Type <RETURN> to continue... " ?___

6. The <RETURN> key clears the computer screen, then triggers

the display of all graphic data files currently existing on the hard disc. The file directory information is followed by:

"What file name contains your data?"
"File designation must be dta (ie); filename.dta "

"Filename.dta ..." ?___

A correct filename input starts {DAMPCALC.BAS} execution which is confirmed by the screen prompt:

"Loading the contents of " (filename) "into the computer..."

"Disc file contents are ... Please wait... "

7. While it loads the file, the computer echoes the file contents to the screen. This exhibition of {DAMPCALC.BAS} program execution is also practiced for subsequent program sections. Screen messages appear that are descriptive of the program operations being executed. The operator may use the <Ctrl>(NumLck> keys to temporarily stop the screen displays for examination. Program execution starts again with any key pressed.

8. The next screen message requesting operator information is:

"Graphic XY-coordinates for ";FILE$;" have been determined. "

"Type <YES> to store XY-coordinates... " ?___

a. Any entry not beginning in "Y" bypasses this option.

b. An affirmative response produces the message:

"What XY-coord filename?"

"File designations must be XYC (ie) FILENAME.XYC "

"Filename.XYC ... " ?___

The operator may enter any filename of less than twelve total characters in length, as long as the designation is "XYC". The S/A Interface program utilizes the "XYC" extension to similarly label all XY coordinate files, making their directory identification much easier. Responding to a valid filename, the program answers:

"Storing your XY-graphic coordinates... "

9. Subsequently, {DAMPCALC.BAS} lists the X and Y coordinates of the selected file to the screen, then requests:

"What SDC/DC correction factor" ?___

During the development of the S/A Interface Program, this researcher chose to write a manual correction factor into the {DAMPCALC.BAS}

:outine. This factor corrects for X and Y coordinates being produced ›y the analyzer and determined by the program in HP-plotter scale 'alues. The factor need be determined only once for each group of lisc files originating from common analyzer screen grids.

a. The method to determine the SDC/DC correction factor is ıccomplished by:

* Use the formulas ...   SDC = 200 PI (W(2) - W(1))/W(n)

DC = (W(2) - W(1))/(2 W(n))

where ... PI = 3.1415927

W(1) & W(2) are the frequency values at the half-power points on either side of the resonant frequency

W(n) is the nodal resonant frequency

* Manually calculate SDC (or DC) from a Signal Analyzer display that has the same coordinate scale as the group of graphic files in question.

* Process the same Signal Analyzer graph used for the SDC (or DC) manual calculation with {DAMPCALC.BAS}, entering an initial SDC/DC correction factor of one (1); (i.e.) no correction, in response to the program request for a factor.

* Divide the manually derived SDC (or DC) value into the {DAMPCALC.BAS} generated SDC produced with no correction. The numerical result of this division is the SDC/DC correction factor, applicable to all graphic files that were produced from common S/A displays. Note that for this factor to be trully common for all files in a group, S/A output voltages, and similar critical configuration set-up page parameters must also be shared.

b. The program algorithm for calculation of SDC and DC epends upon determination of the curve baseline so that half-power oints can be found as (1/square root of 2) times the maximum ampliude. The curve baseline is found by averaging the Y coordinate vales at the left and right ends of the curve in question. HP-plotter oordinate values assigned by the S/A are not zero at the curve miniums, but have magnitudes that must be included when calculating SDC nd DC. See <DAMPCALC.BAS> program lines 3880 - 3900 and GRAPHXYC.BAS> program lines 2200 - 2300.

10. Entering a valid SDC/DC correction factor produces the following screen report:

```
"*******************************************************"

"  VALUES FOR " (filename)   "YMAX(Abs) = " (Ymax coord)

"  SDC = " (SDC) " % "       "X(0) = " (X minimum)

"  DC  = " (DC)          "XMAX(Abs) = " (X maximum)

"-------------------------------------------------------"

" SDC/DC CORRECTION FACTOR OF " (correction factor)

"*******************************************************"

"Type any key + <RETURN> to ReEnter the SDC/DC factor "
```

The listed YMAX(Abs) value is the maximum vertical graphic coordinate, or the maximum amplitude in HP-plotter scale terms of the curve in question. X(0) is the minimum horizontal graphic coordinate, while XMAX(Abs) is the maximum value in HP-plotter scale terms for the {DAMPCALC.BAS} processed curve.

11. In the same fashion as prior discussed sub-programs, this routine has a final screen message which enables {DAMPCALC.BAS} to be rerun for a different disc file.

"Type yes to examine another disc file... " ?___

Any response not starting with "Y" returns the main menu to the computer screen.

K. GRAPH XY COORDINATES {GRAPHXYC.BAS} The last main menu selection in the S/A Interface Program is a routine that graphically displays the curves stored to disc as X and Y coordinates by {DAMPCALC.BAS}, the previous menu selection. {GRAPHXYC.BAS} reads the files with ".XYC" extensions from the hard disc; finds the maximum amplitude coordinate pair; "smoothes" the curve coordinates and finds the half-power points; calculates the Specific Damping Capacity (SDC) and Damping Coefficient; and then displays the selected graph on the computer screen with the half-power points marked. The original curve and its "smoothed" version are both displayed to provide verification that the processed graph is a close approximation of the original.

1. Main menu selection 10 executes {GRAPHXYC.BAS} which begins with the banner instructions:

```
***************** Program <GRAPHXYC.BAS> *****************************
** This program loads the XY-coordinate contents of a designated  **
** disc file into the computer, then displays those coordinates   **
** graphically on the computer screen.  A second curve is also    **
** displayed that is a SMOOTHED version of the disc file graph.   **
** The  SMOOTHED  version is used for SDC & DC calculations, so   **
** its visual FIT to the disc file graphic data is displayed on   **
** the computer screen for comparison.  XY-coord disc files used  **
** by this program must have been produced by <DAMPCALC.BAS>.     **
********************************************************************
** Note: Disc files for this program must have file designations  **
** of  <XYC>  (ie)  FILENAME.XYC    Other file types will not     **
** load correctly.                                                **
********************************************************************
```

"Type <RETURN> to continue... " ?___

2. Typing the <RETURN> key clears the screen, displays the directory of XY-coordinate files, and exhibits the program request for a filename:

"What file name contains your XY-coord data?"
"File designation must be xyc (ie); filename.xyc "

"Filename.xyc ..." ?___

Only XY-coordinate files generated by {DAMPCALC.BAS}, or files that have the X and Y coordinates stored as integers in the sequence "X1 Y1 X2 Y2 X3 Y3 X4 Y4 etc. ..." will load correctly. Upon the entry of a valid filename, the routine advises:

"Loading the contents of " (filename) " into the computer... "
"Disc file contents are ... Please wait... "

```
023   19  1087    21  1151    21  1215    18  1279    20  1343    21
404   20  1471    21  1535    22  1599    24  1663    26  1727    27
855   27  1919    29  1983    30  2047    30  2111    31  2175    32
239   32  2303    33  2367    35  2431    38  2559    38  2623    40
687   41  2751    42  2815    44  2943    44  3007    46  3071    46
135   48  3199    50  3263    50  3327    52  3391    53  3455    56
519   58  3583    61  3647    61  3711    63  3775    63  3839    64
903   65  3967    70  4031    71  4095    71  4159    74  4223    75
287   78  4351    82  4415    85  4479    83  4543    84  4607    90
671   92  4735    91  4799    92  4863    94  4927    95  4991    97
055  103  5119   103  5183    99  5247    99  5311    94  5375    93
....etc. ...........
```

imilar to prior programs, the disc file contents are exhibited on the creen for verification of correct operation. Note that, depending on he coordinate values, file contents may not line up in neat columns s the above example implies.

3. {GRAPHXYC.BAS} also calculates SDC and DC for the chosen raph. The calculation also depends on the entry of an SDC/DC correc-

tion factor entered by the operator, just as it did in {DAMPCALC.BAS}. The procedure to calculate the correction factor is exactly the same as that explained in paragraph J above. Therefore, it is not reiterated here.

a. After entry of the SDC/DC correction factor, the results appear as:

```
"****************************************"

      "RESULTS FOR " (filename)

      "SDC = " (SDC) " Percent "

      "DC  = " (DC)

  "PLOTTER COORD YMAX(Abs) = " (Y maximum)

  "X(0) = " (X minimum)  "X(MAX) = " (X maximum)

  "SDC/DC CORRECTION FACTOR IS " (factor)

"****************************************"

 "Type any key + <RETURN> to ReEnter the SDC/DC factor " ?___
```

4. Pressing the <RETURN> key only (entering a null response) will advance program execution to the prompt:

```
"Type  <RETURN>  for graph of file data." ?___
```

Typing the <RETURN> key once again will clear the screen and graph the curve on the computer monitor. The graph is displayed with a ten-by-ten grid across the screen. In the upper left corner of the graphic display, the message appears:

```
"SCREEN TRACE OF " (filename) " SIGNAL ANALYZER DISC FILE"
"A positive XFCTR moves trace to the right. A positive YFCTR moves "
"screen trace down.  <XFCTR + YFCTR = 0> exits the graph mode ... "
"WHAT XFCTR = " ?___
"WHAT YFCTR = " ?___
```

a. The computer monitor graphic origin is at the upper left of the monitor screen. Thus, monitor X-coordinates increase in value from left to right, the usual convention. But, Y-coordinates increase in value on the monitor from top to bottom, the reverse of the common convention.

b. To move the graphic curve LEFT and DOWN, enter a negative integer for "XFCTR" and a positive integer for "YFCTR". Integer multiples of twenty (20) to thirty (30) provide reasonable motion in the desired directions.

5. The graphic display section of {GRAPHXYC.BAS} can be exited ith two successive null entries [<RETURN> and <RETURN>], or any two alues XFCTR + YFCTR which sum to zero. That combination produces the inal screen message:

"Type yes to examine another disc file... " ?___

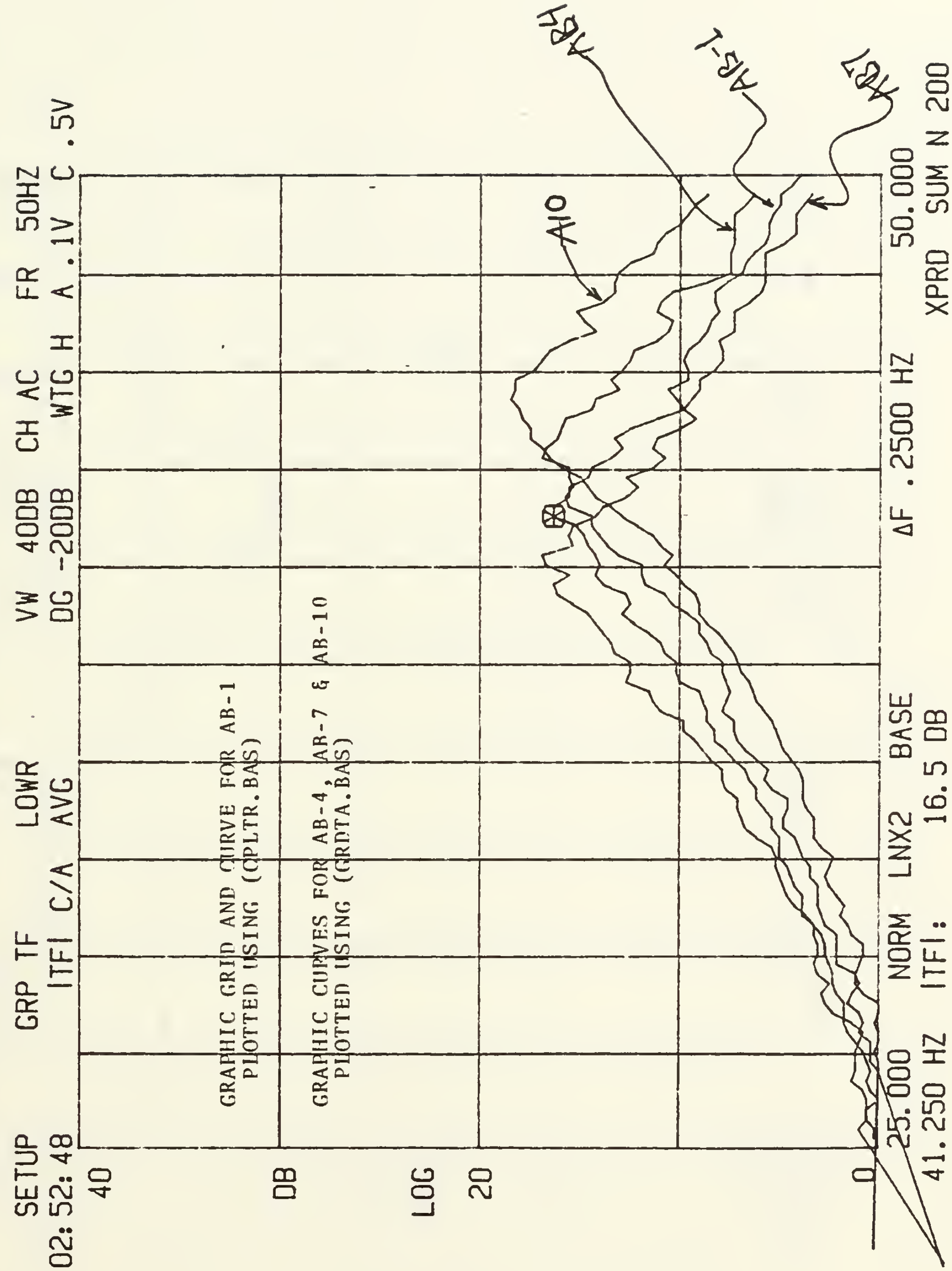
negative answer (not beginning with "Y") returns the operator to the ain menu.

6. The curve smoothing routine used in <GRAPHXY.BAS> is listed n program lines 1420 - 2180. The smoothing routine operates on the -coordinate values only; X-coordinate values and indecis remain un-hanged. The routine calculates an average Y-coordinate increment for he left half of the maxima curve (LYI). Similarly, an average Y-oordinate increment is figured for the curve portion to the right of ts maximum (RYI). These average increments are proportionally summed n program lines 1980 and 2140 to approximate the left and right hal-es of the original curve with more continuous versions. Factors of .05 and 1.2 appearing in these program lines can be changed by the rogrammer to vary the shape of the "smoothed" curves produced. These actor values worked well for modal maximums displayed over a 25 Hz requency range.

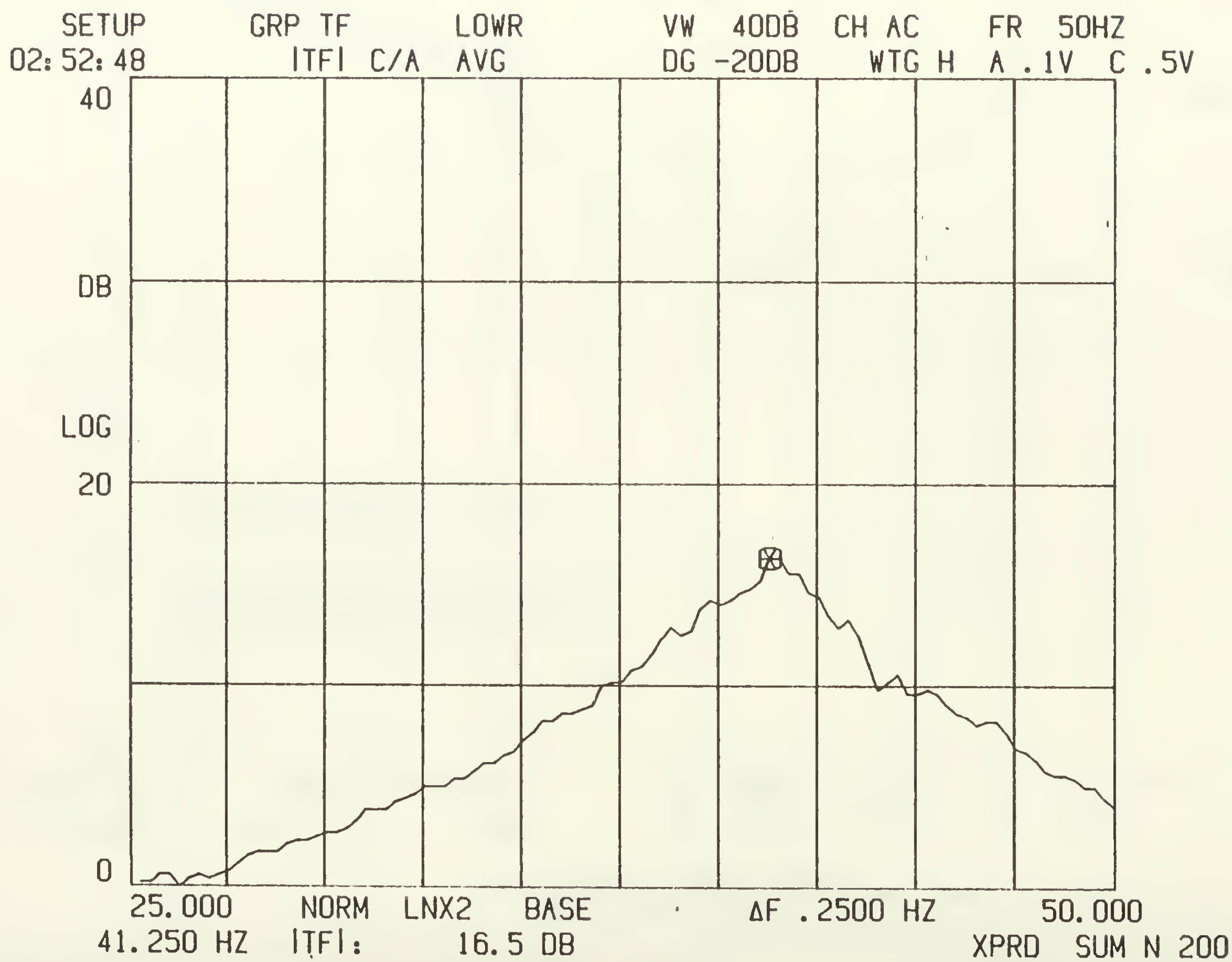
APPENDIX C

GRAPHIC PROGRAM OUTPUT/RESULTS

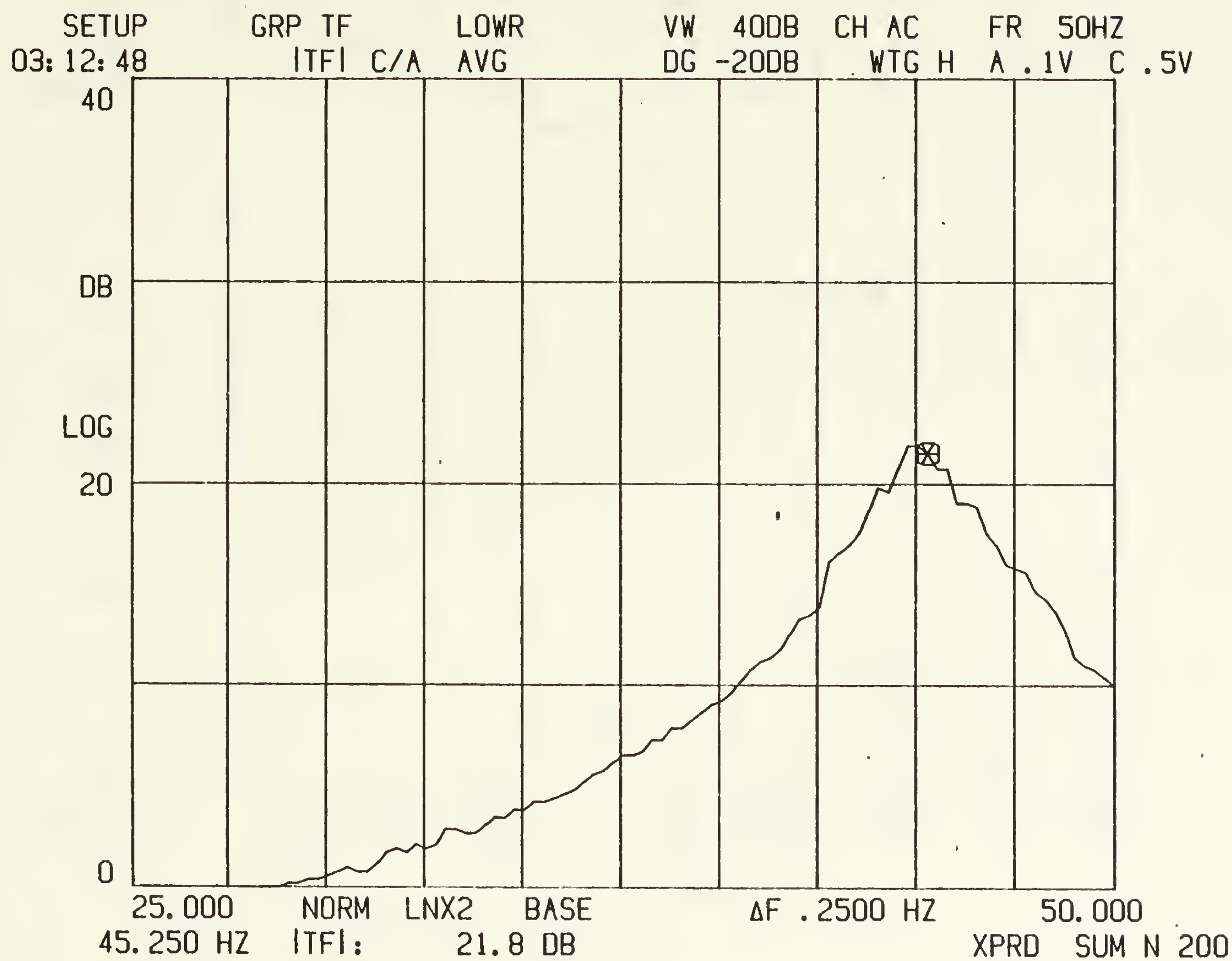
COINCIDENT GRAPHIC PLOTS FOR SAMPLES
AB-1, AB-4, AB-7 & AB-10



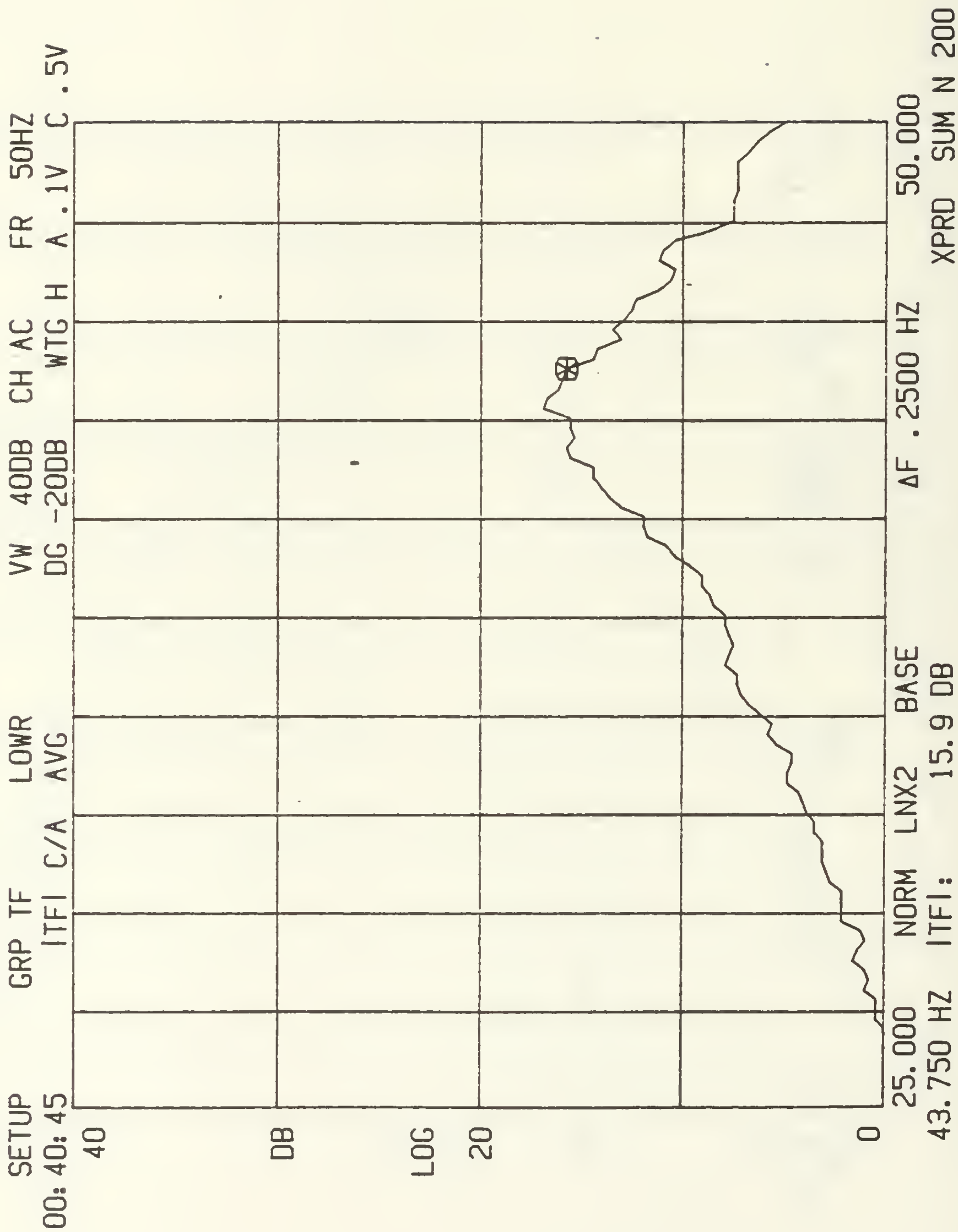
AB-1 1100 Deg C & FC



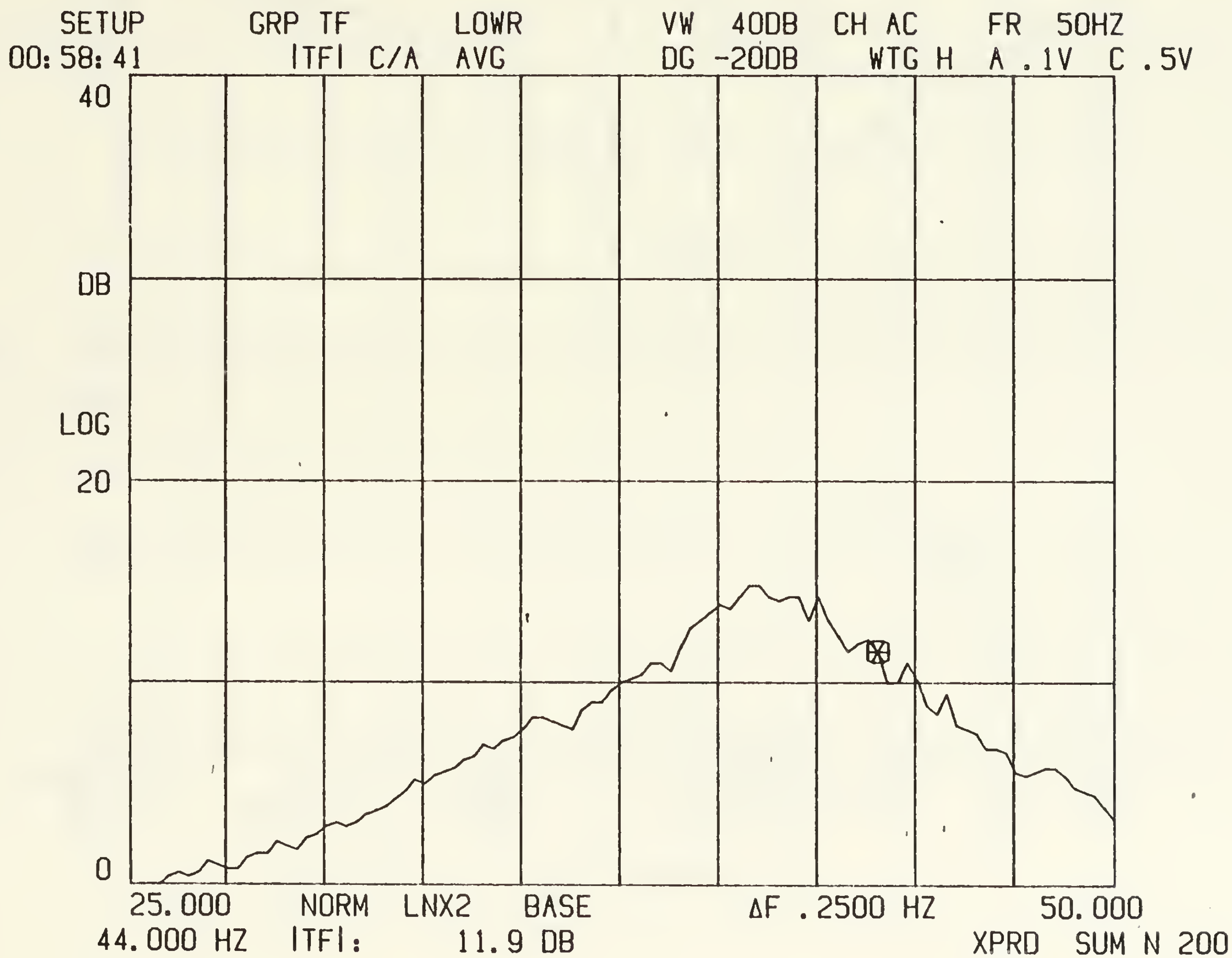
AB-2 1100 Deg C & FC



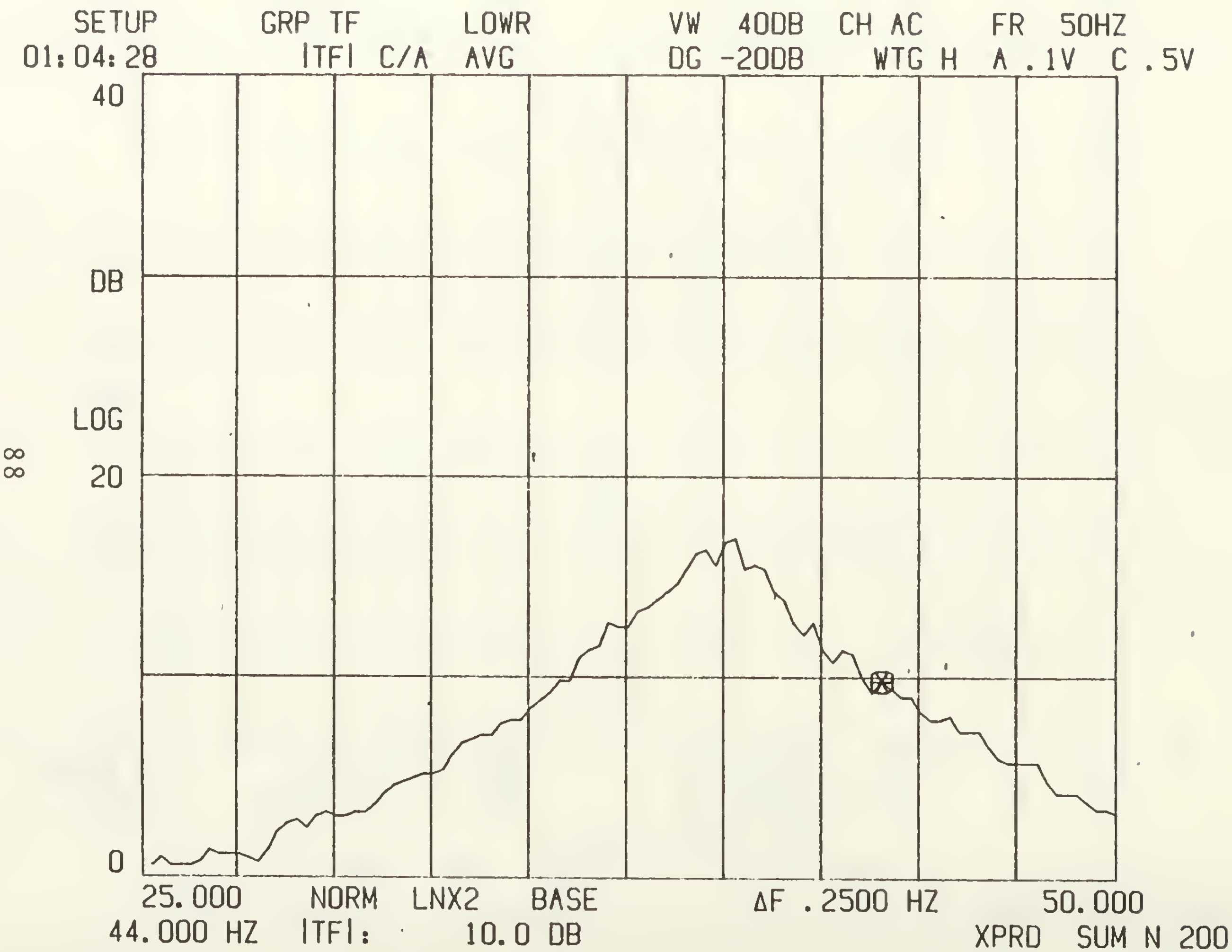
AB-4 1050 Deg C & FC



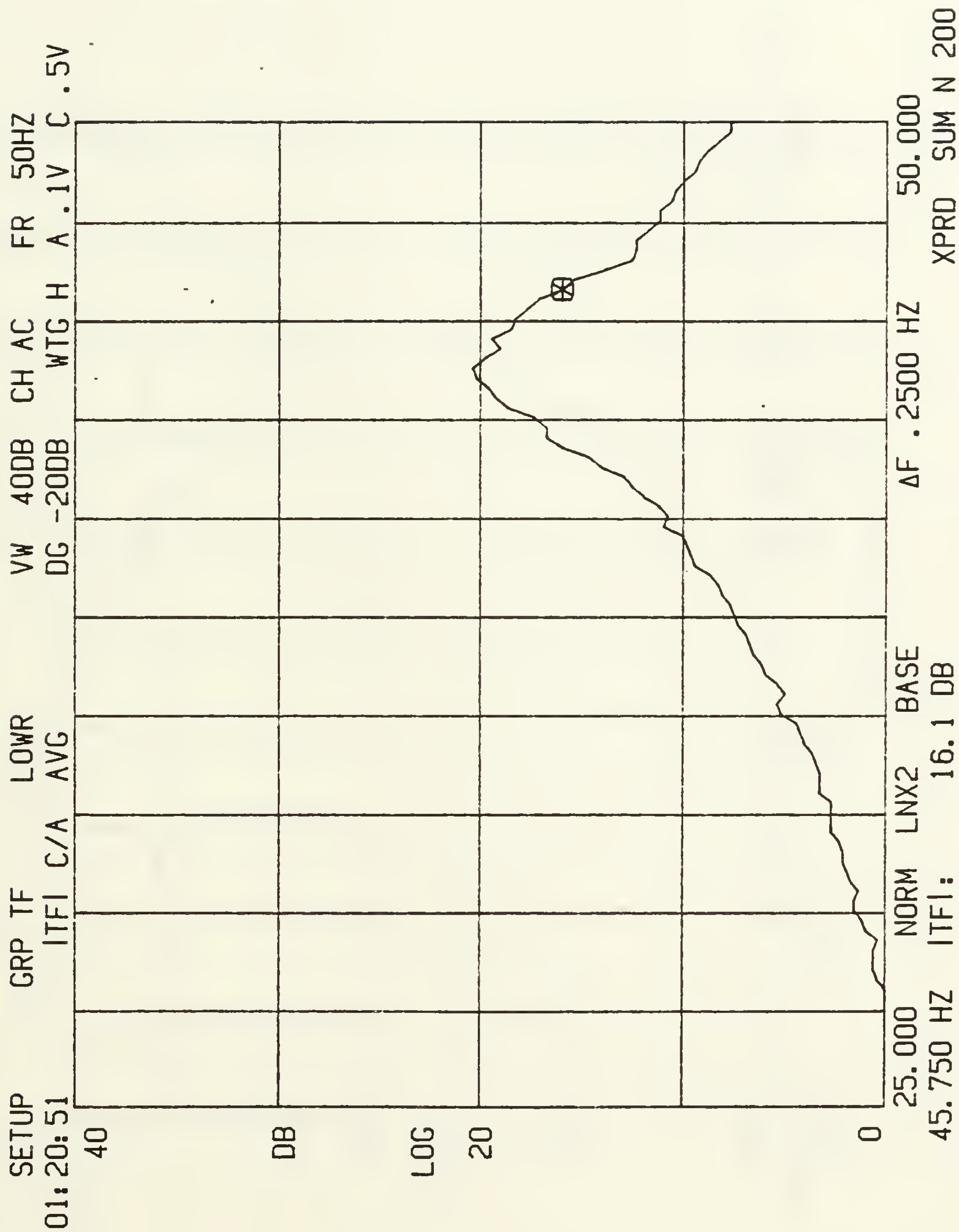
AB-5 1050 Deg C & FC



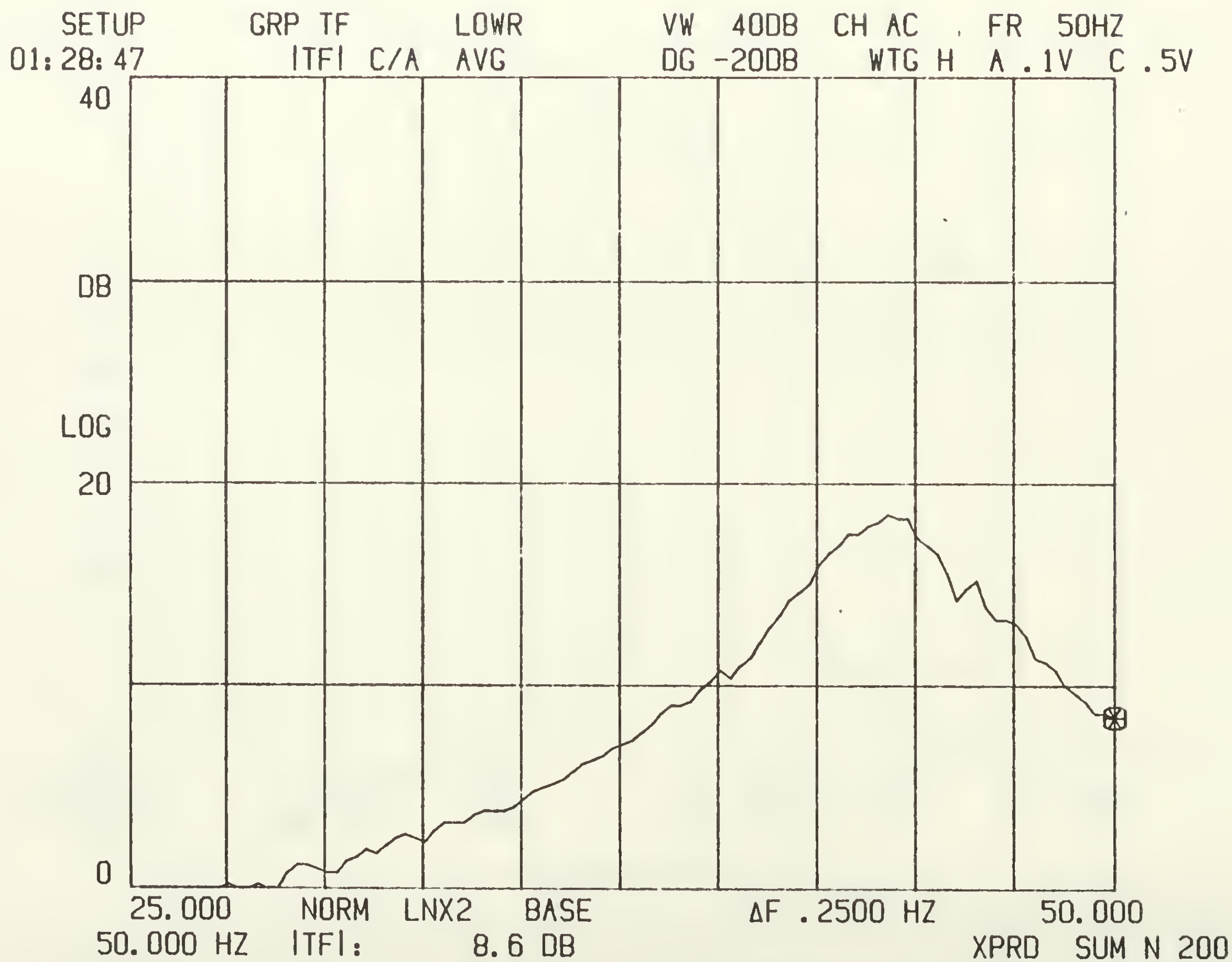
AB-7 1000 Deg C & FC



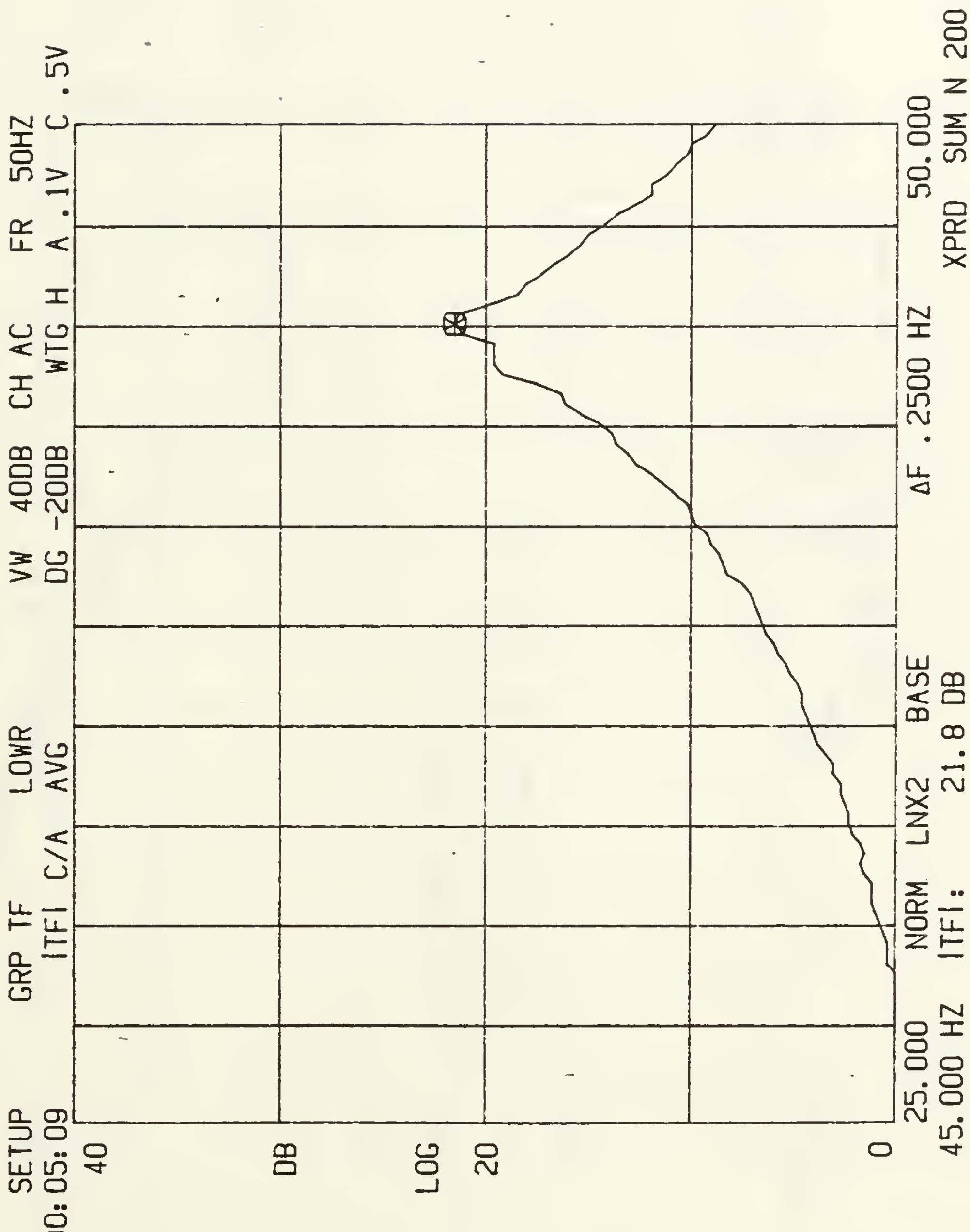
AB-8 1000 Deg C & FC



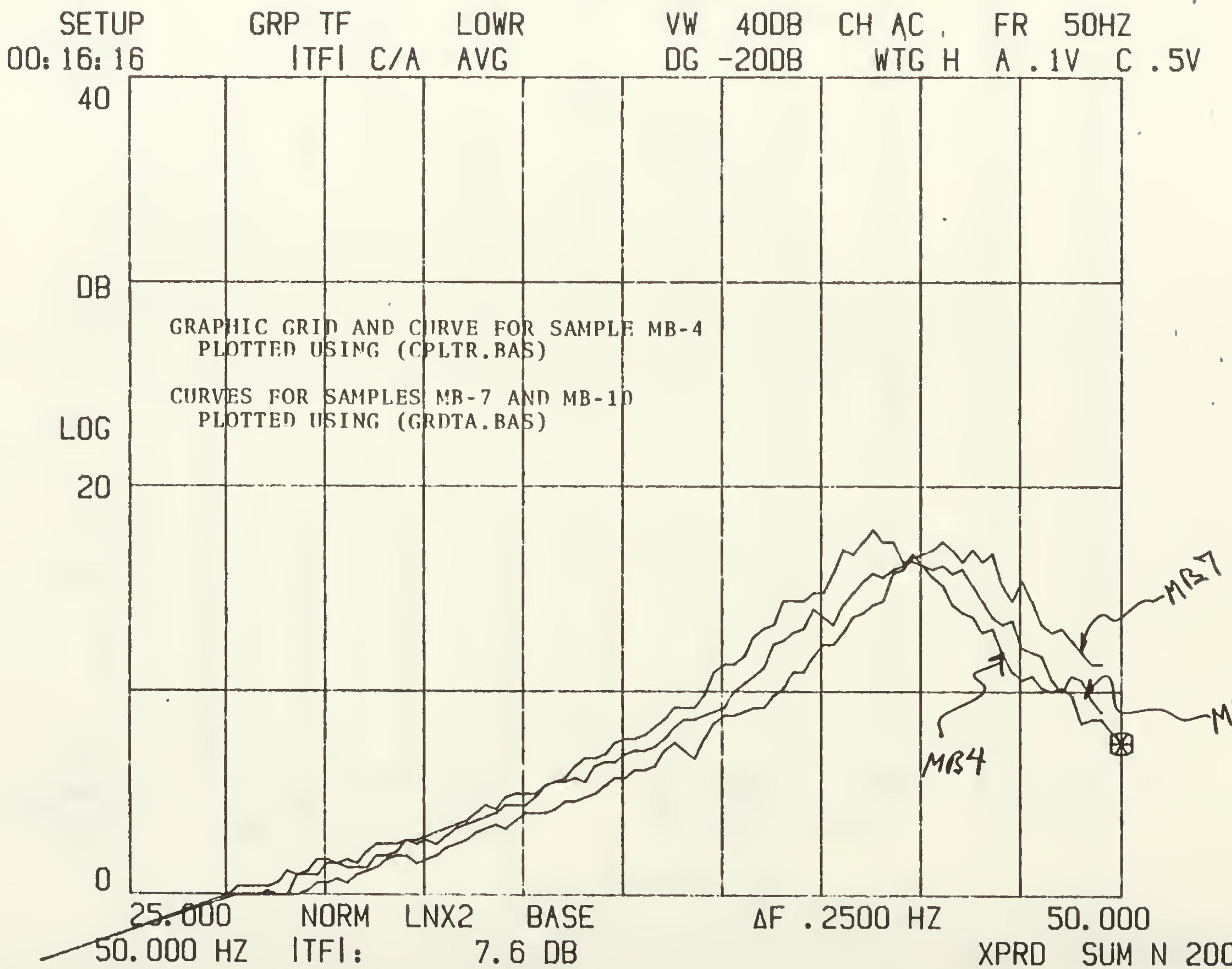
AB-10 950 Deg C & FC



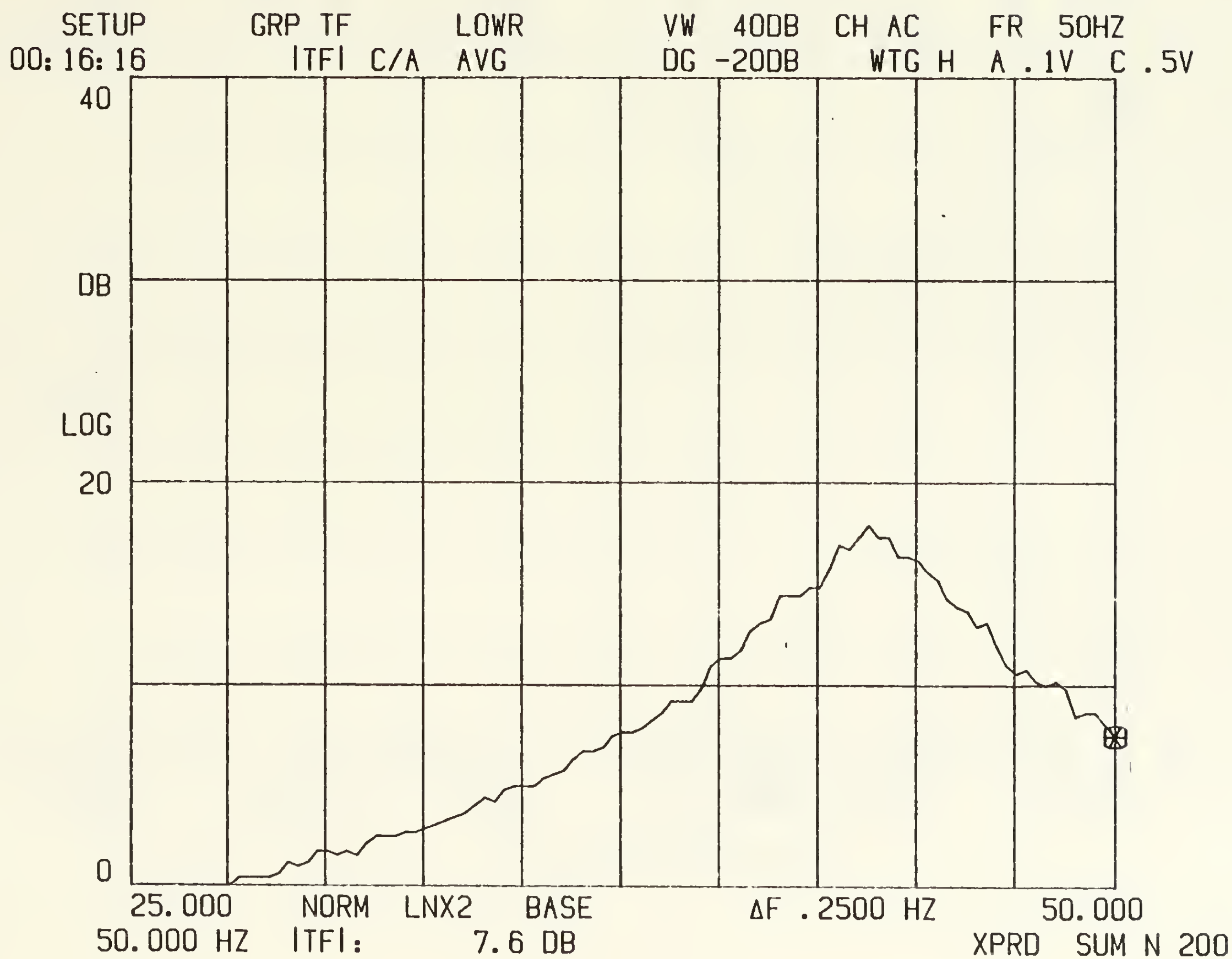
AB-11 950 Deg C & FC



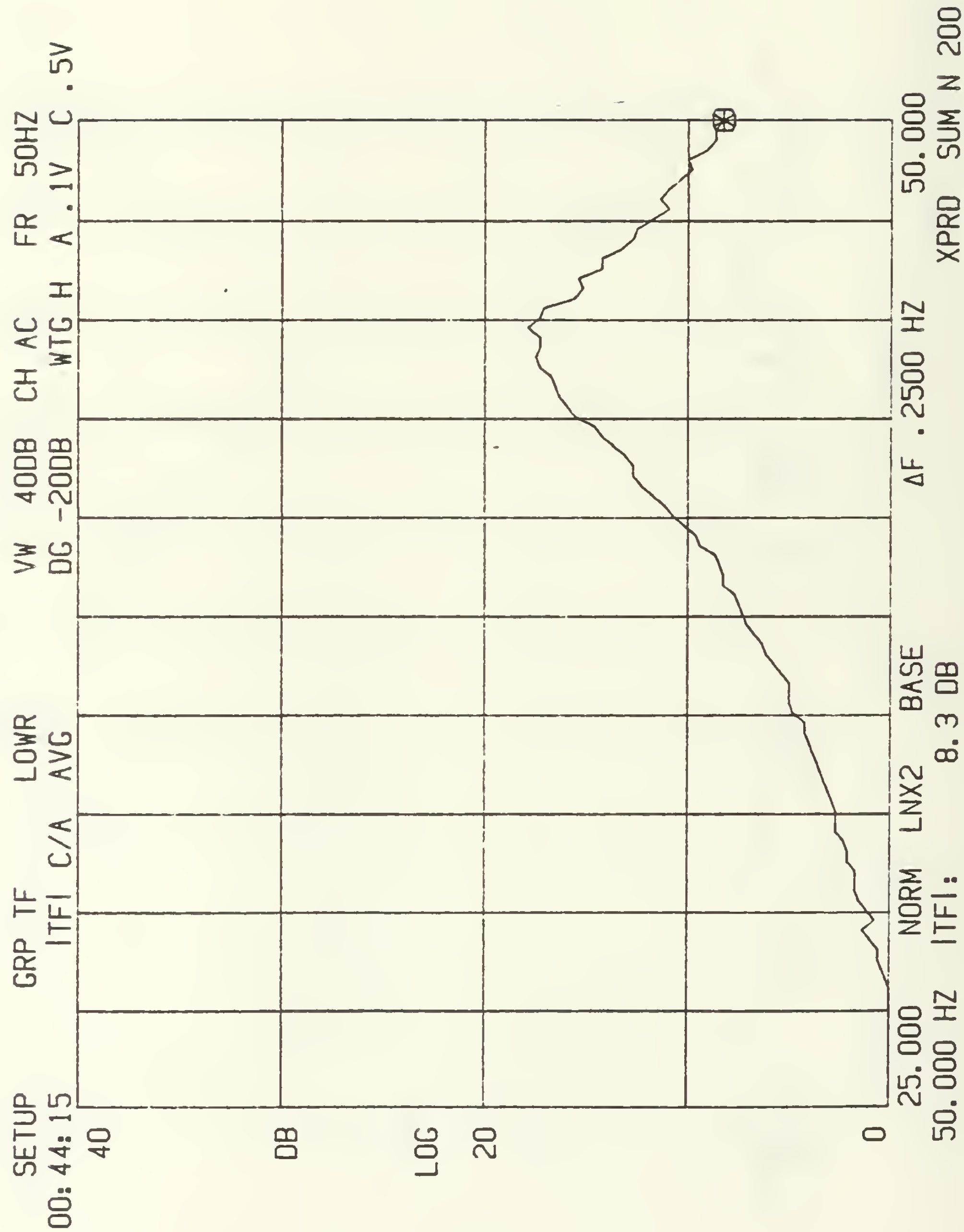
COINCIDENT GRAPHIC PLOTS FOR SAMPLES
MB-4, MB-7 & MB-10



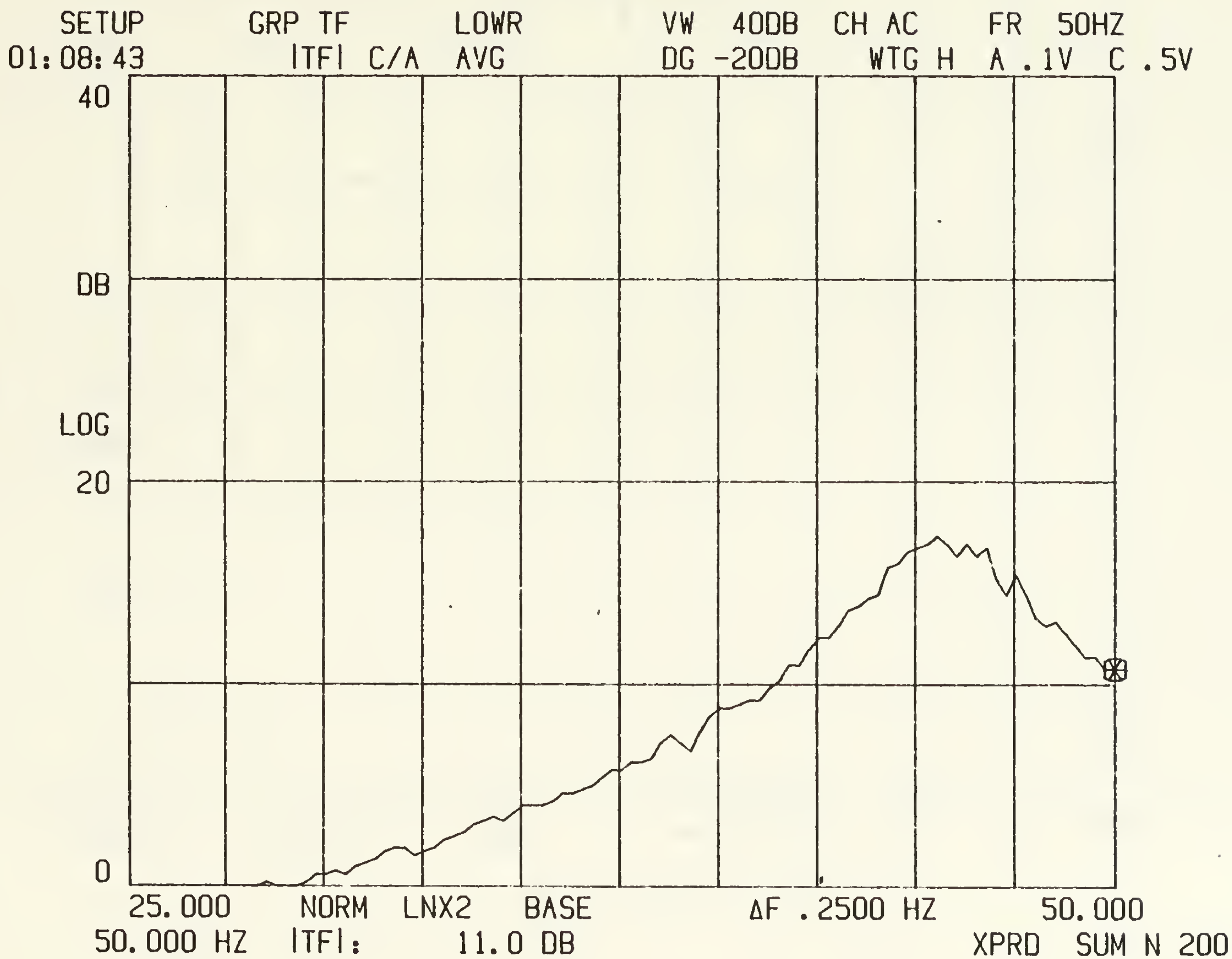
MB-4 1050 Deg C & FC



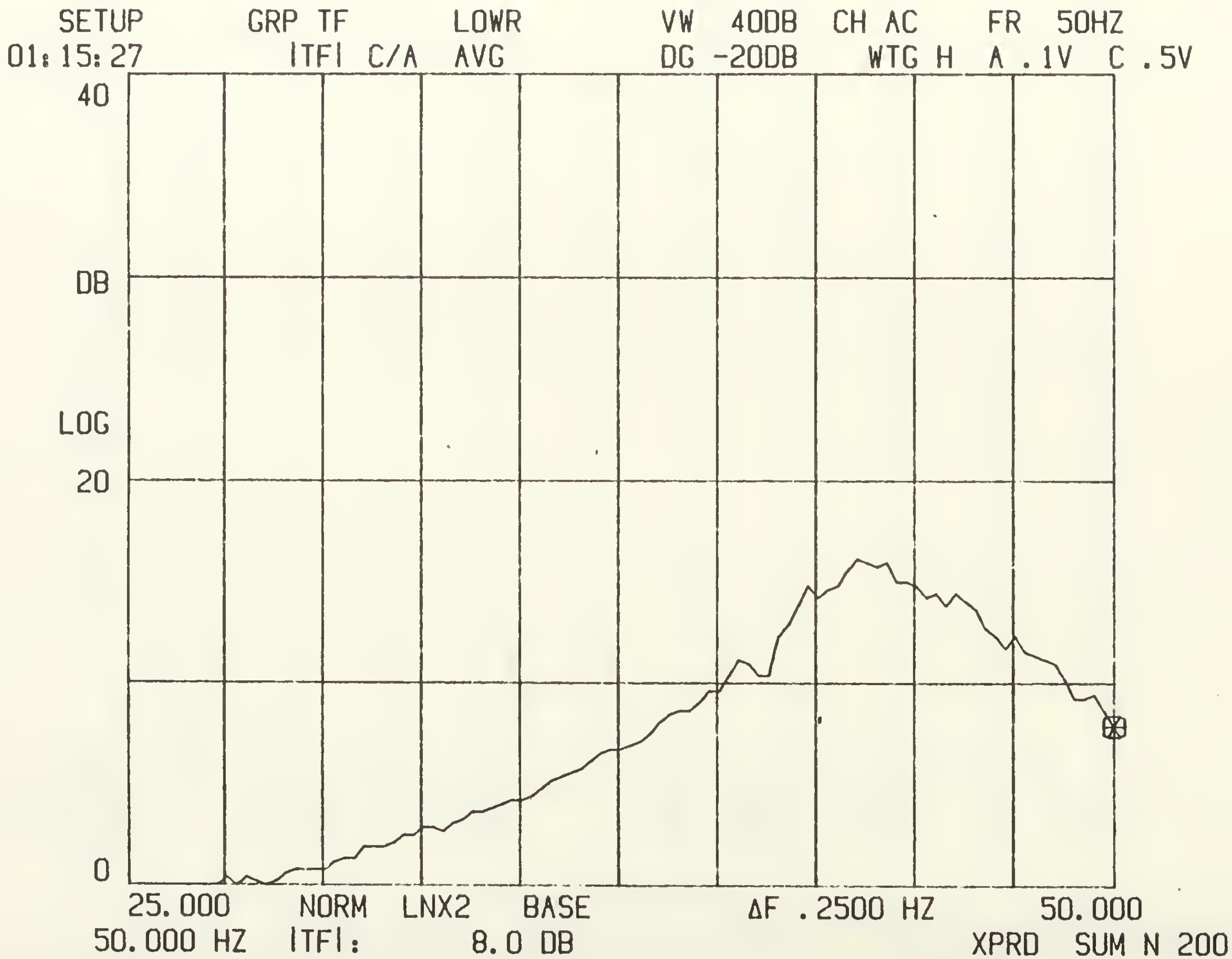
MB-5 1050 Deg C & FC



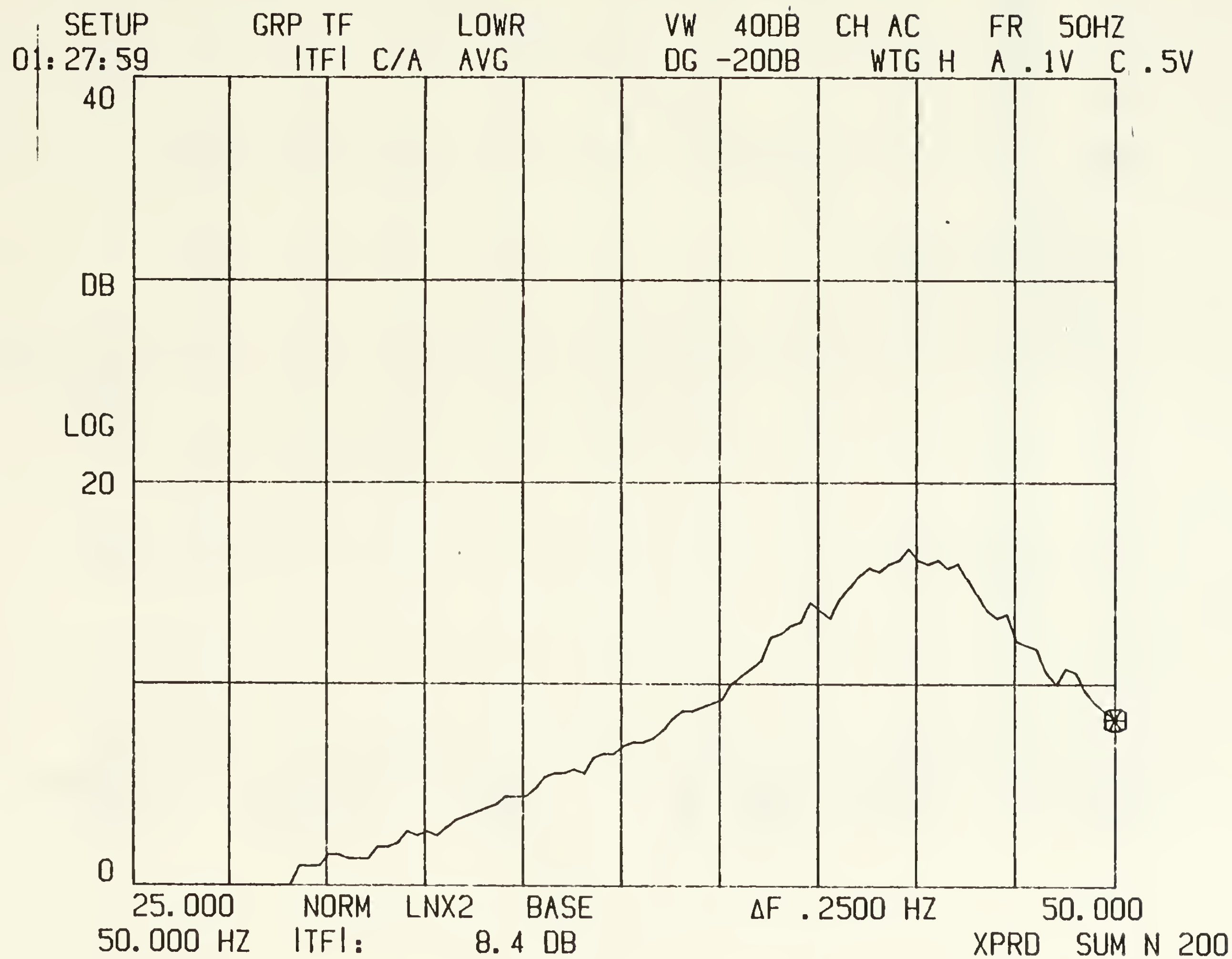
MB-7 1000 Deg C & FC



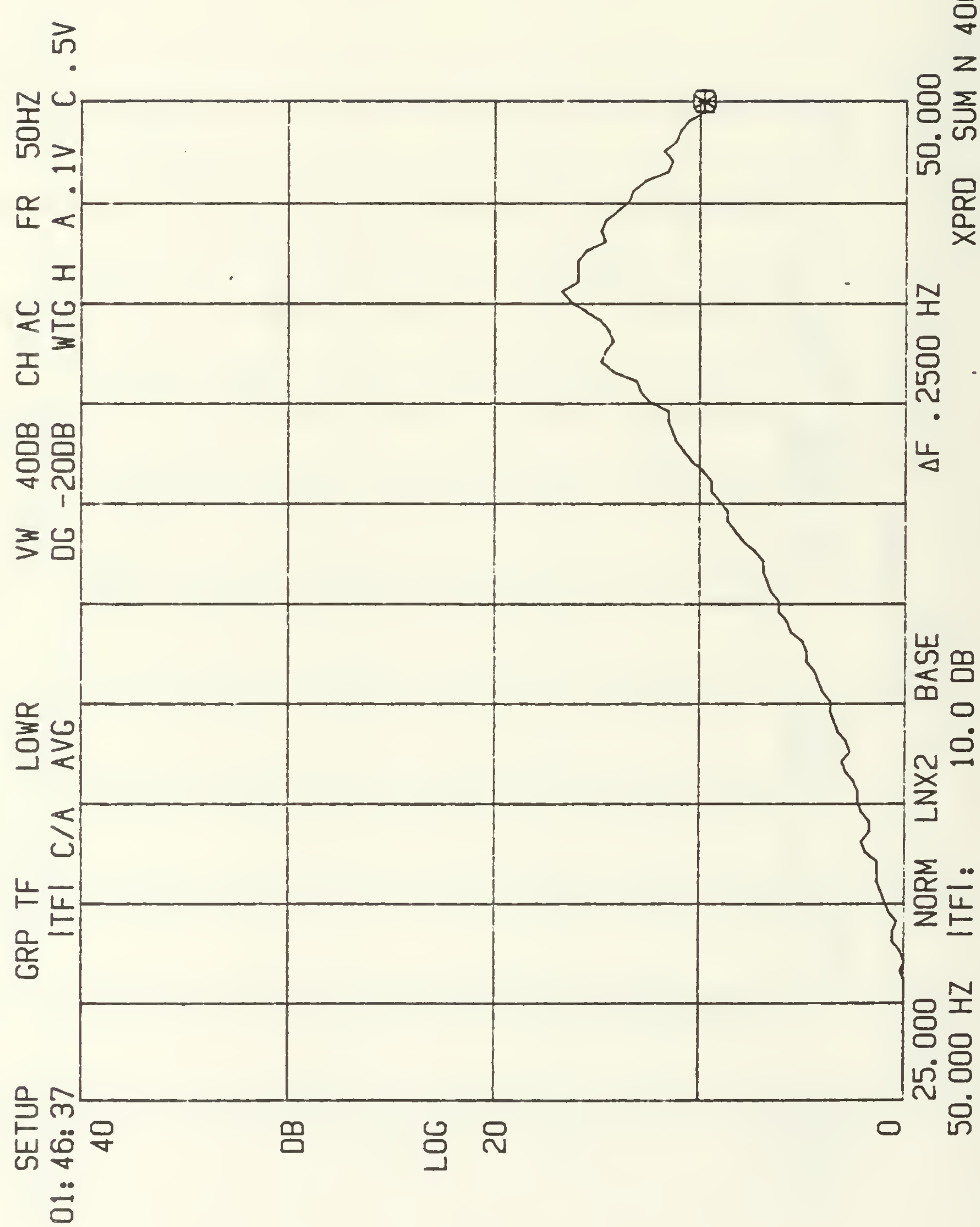
MB-8 1000 Deg C & FC



MB-10 950 Deg C & FC



MB-11 950 Peg C & FC



APPENDIX D

NUMERICAL SDC AND DC PROGRAM RESULTS

```
******************************************
      RESULTS FOR a1fc1100.xyc
      SDC =  62.19728  Percent
      DC  =  4.949503E-02
  PLOTTER COORD YMAX(Abs) =  103
  X(0) =  1023   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.6523
******************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN  3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

```
******************************************
      RESULTS FOR a2fc1100.xyc
      SDC =  40.32938  Percent
      DC  =  .0320931
  PLOTTER COORD YMAX(Abs) =  132
  X(0) =  1727   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.6523
******************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN  3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

```
*******************************************
      RESULTS FOR a4fc1050.xyc
      SDC =  52.9629  Percent
      DC  =  4.214654E-02
  PLOTTER COORD YMAX(Abs) =  106
  X(0) =  1407   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.6523
*******************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN  3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

```
*******************************************
      RESULTS FOR a5fc1050.xyc
      SDC =  66.83013  Percent
      DC  =  5.318173E-02
  PLOTTER COORD YMAX(Abs) =  95
  X(0) =  1087   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.6523
*******************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN  3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

```
******************************************

      RESULTS FOR a7fc1000.xyc

      SDC =  66.34463  Percent

      DC  =  5.279538E-02

  PLOTTER COORD YMAX(Abs) =  106

  X(0) =  1023   X(MAX) =  7231

  SDC/DC CORRECTION FACTOR IS  2.6523

******************************************

Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN  3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN



******************************************

      RESULTS FOR a8fc1000.xyc

      SDC =  50.58213  Percent

      DC  =  4.025198E-02

  PLOTTER COORD YMAX(Abs) =  124

  X(0) =  1471   X(MAX) =  7231

  SDC/DC CORRECTION FACTOR IS  2.6523

******************************************

Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN  3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

```
******************************************
      RESULTS FOR a10fc950.xyc
      SDC =  46.86655  Percent
      DC  =  3.729521E-02
  PLOTTER COORD YMAX(Abs) =  114
  X(0) =  1407   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.6523
******************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor

1LIST   2RUN   3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

```
******************************************
      RESULTS FOR a11fc950.xyc
      SDC =  42.84997  Percent
      DC  =  3.409892E-02
  PLOTTER COORD YMAX(Abs) =  130
  X(0) =  1535   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.6523
******************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor

1LIST   2RUN   3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

```
******************************************
      RESULTS FOR m4fc1050.xyc
      SDC =  62.94721  Percent
      DC  =  .0500918
  PLOTTER COORD YMAX(Abs) =  111
  X(0) =  1471   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.01912
******************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN   3LOAD"   4SAVE"   5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN

******************************************
      RESULTS FOR m5fc1050.xyc
      SDC =  60.23935  Percent
      DC  =  4.793695E-02
  PLOTTER COORD YMAX(Abs) =  111
  X(0) =  1343   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.01912
******************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN   3LOAD"   4SAVE"   5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

```
*****************************************
      RESULTS FOR m7fc1000.xyc
      SDC =  51.8725  Percent
      DC  =  4.127883E-02
  PLOTTER COORD YMAX(Abs) =  108
  X(0) =  1535   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.01912
*****************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN  3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN



*****************************************
      RESULTS FOR m8fc1000.xyc
      SDC =  63.66265  Percent
      DC  =  5.066113E-02
  PLOTTER COORD YMAX(Abs) =  102
  X(0) =  1407   X(MAX) =  7231
  SDC/DC CORRECTION FACTOR IS  2.01912
*****************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor


1LIST   2RUN  3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

```
*****************************************
     RESULTS FOR m10fc950.xyc
     SDC =  53.54609  Percent
     DC  =  4.261063E-02
 PLOTTER COORD YMAX(Abs) =  105
 X(0) =  1599   X(MAX) =  7231
 SDC/DC CORRECTION FACTOR IS  2.01912
*****************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor

1LIST   2RUN   3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN

*****************************************
     RESULTS FOR m11fc950.xyc
     SDC =  55.69478  Percent
     DC  =  .0443205
 PLOTTER COORD YMAX(Abs) =  105
 X(0) =  1599   X(MAX) =  7231
 SDC/DC CORRECTION FACTOR IS  2.01912
*****************************************
Type any key + <RETURN> to ReEnter the SDC/DC factor

1LIST   2RUN   3LOAD"  4SAVE"  5CONT 6,"LPT1 7TRON 8TROFF9EXIT   0SCREEN
```

APPENDIX E

SIGNAL ANALYZER INTERFACE PROGRAM CODE

```
100 REM ***************** <MENU.BAS> PROGRAM ******************************
120 REM This is the MAIN MENU for all the Signal Analyzer utility programs.
140        CLS : BEEP : BEEP : REM Clear screen.
160 PRINT "   **************************************************************"
180 PRINT "   **            SIGNAL ANALYZER INTERFACE PROGRAM               **"
200 PRINT "   **                    MAIN OPTIONS MENU                       **"
220 PRINT "   **                                                            **"
240 PRINT "   **    <MCNFG.BAS> ........................  1                 **"
260 PRINT "   **    <FPLOT.BAS> ........................  2                 **"
280 PRINT "   **    <PRPLOT.BAS> .......................  3                 **"
300 PRINT "   **    <CPLTR.BAS> ........................  4                 **"
320 PRINT "   **    <INTDTA.BAS> .......................  5                 **"
340 PRINT "   **    <GRDTA.BAS> ........................  6                 **"
360 PRINT "   **    <SCRNDTA.BAS> ......................  7                 **"
380 PRINT "   **    <ITERPLOT.BAS> .....................  8                 **"
400 PRINT "   **    <DAMPCALC.BAS> .....................  9                 **"
420 PRINT "   **    <GRAPHXYC.BAS> ..................... 10                 **"
440 PRINT "   **    EXIT THIS MENU ..................ANY OTHER KEY          **"
460 PRINT "   **                                                            **"
480 PRINT "   **                                                            **"
500 PRINT "   **                                                            **"
520 PRINT "   **                                                            **"
540 PRINT "   **    PERKINS/PATCH MASTERS THESIS (NAVPGSCOL)                **"
560 PRINT "   **    COPYRIGHT AUGUST 20, 1987 UNITED STATES NAVY            **"
580 PRINT "   **************************************************************"
600 LOCATE 18, 18 : INPUT "CHOICE ";ANS$
620      LOCATE 18,30 : PRINT "OK!" : LOCATE 18,35
640    IF ANS$ = "1" THEN RUN "MCNFG"
660    IF ANS$ = "2" THEN RUN "FPLOT"
680    IF ANS$ = "3" THEN RUN "PRPLOT"
700    IF ANS$ = "4" THEN RUN "CPLTR"
720    IF ANS$ = "5" THEN RUN "INTDTA"
740    IF ANS$ = "6" THEN RUN "GRDTA"
760    IF ANS$ = "7" THEN RUN "SCRNDTA"
780    IF ANS$ = "8" THEN RUN "ITERPLOT"
800    IF ANS$ = "9" THEN RUN "DAMPCALC"
820    IF ANS$ = "10" THEN RUN "GRAPHXYC"
840     CLS : PRINT : PRINT "      EXITING MAIN MENU... "
860 END
```

```
100 REM **************** gpib-pc lead-in program lines **********************
120        CLEAR   ,59504!:REM   BASIC Declarations
140        IBINIT1 = 59504!
160        IBINIT2 = IBINIT1 + 3:REM Lines 120 through 220 MUST be included in
180        BLOAD "bib.m",IBINIT1
200        CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,
           IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,
           IBWRTF)
220        CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,
           IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,IBWRTI,IBRDIA,
           IBWRTIA,IBSTA%,IBERR%,IBCNT%)
240 REM **************** gpib-pc lead-in program lines end here *************
260   KEY 9, "EXIT" : ON KEY (9) GOSUB 1740 : KEY (9) ON`: REM Interrupt trap.
280 REM    ----  program Machine Configuration <MCNFG.BAS> ----
300 DIM A%(1000)
320 CLS : SOUND 900, 8 : SOUND 880, 6
340 PRINT "********* Machine Configuration Program <MCNFG.BAS> **************"
360 PRINT "** This program can store signal analyzer machine configuration **"
380 PRINT "** codes obtained originally from the analyzer itself.  The      **"
400 PRINT "** program can also set up the signal analyzer, using any        **"
420 PRINT "** configuration data files previously saved to computer disc.   **"
440 PRINT "*****************************************************************"
460 PRINT "**   Ensure that the Signal Analyzer is connected and  ON        **"
480 PRINT "**   and that its GPIB address is 25.                            **"
500 PRINT "*****************************************************************"
520 DEV$ = "DEV1" : CNT% = 1200 : AT$ = " FPKEY 8,47 " : CMD1$ = "mcnfg?"
540  CMD2$ = "mcnfg "
560 PRINT : PRINT "Type  1  to obtain & store the current configuration.  "
580 PRINT  "Type  2  to set up analyzer with an existing config'tn file. "
600 PRINT: INPUT " 1 or 2 ... "; ANS$
620 REM Following three lines are traps for invalid/valid keyboard responses.
640     IF ANS$ = "1" THEN 720
660     IF ANS$ = "2" THEN 1120
680        GOTO 320
700 REM ***** start program section to obtain & store analyzer config'tn ******
720 CALL IBFIND (DEV$,DV%)
740 CALL IBCMD (DV%, AT$ )
760 CALL IBWRT (DV%, CMD1$)
780 CALL IBRDI (DV%, A%(1), CNT%)
800 CLS: PRINT : PRINT "This is a sample of the analyzer config'tn data ... "
820 FOR I = 1 TO 400: PRINT A%(I);: NEXT I
840 PRINT : PRINT "What file name for disc storage? "
860 SOUND 900, 8 : SOUND 880, 6
880 PRINT "Note: The file designation must be  cfg  (ie); filename.cfg "
900 PRINT "Type  <RETURN>  to bypass disc storage. "
920 PRINT : INPUT "Filename.cfg ...";FILE$
940     IF FILE$ = "" THEN 1580
960     IF LEN(FILE$) > 12 THEN 840 : REM Limits filename length to < 12
980     IF RIGHT$(FILE$,3) = "CFG" THEN 1020
1000     IF RIGHT$(FILE$,3) <> "cfg" THEN 840
1020 CLS: PRINT : PRINT "Storing analyzer config'tn data under filename ";FILE$
1040 PRINT : PRINT "Please wait... "
1060 OPEN "c:\gpib-pc\" + FILE$ FOR OUTPUT AS #1
1080 FOR J = 1 TO 600 : PRINT #1, USING "########"; A%(J); : NEXT J
1100 CLOSE #1 : GOTO 1580
```

```
1120 REM ********* start of program section to use existing disc file *********
1140     ON ERROR GOTO 1700 : CLS : REM No file error trap & clear screen.
1160 PRINT : FILES "*.cfg" : SOUND 900, 8 : SOUND 880, 6
1180 PRINT : PRINT "What file name for retrieval from disc?"
1200 INPUT "Note: The file designation must be cfg (ie); filename.cfg "; FILE$
1220     IF RIGHT$(FILE$,3) = "CFG" THEN 1260
1240     IF RIGHT$(FILE$,3) <> "cfg" THEN 1160
1260 PRINT : PRINT "Opening file ";FILE$;" ...Please wait. "
1280 OPEN "c:\gpib-pc\" + FILE$ FOR INPUT AS #1
1300     FOR J = 1 TO 600 : IF EOF(1) THEN 1320 : INPUT #1, A%(J) : NEXT J
1320 CLOSE #1 :CLS :PRINT : PRINT "Sample contents of file ";FILE$;" are ... "
1340     FOR I = 1 TO 400 : PRINT A%(I); : NEXT I
1360 PRINT : PRINT : SOUND 900, 8 : SOUND 880, 6
1380 INPUT "Type  yes  to set up signal analyzer using this file. ";ANS$
1400     IF LEFT$(ANS$,1) = "y" THEN 1440
1420     IF LEFT$(ANS$,1) <> "Y" THEN 1580
1440 CALL IBFIND (DEV$, DV%)
1460 CALL IBCMD (DV%, AT$)
1480 CALL IBWRT (DV%, CMD2$)
1500 CALL IBWRTI (DV%, A%(1), CNT%)
1520 PRINT : INPUT "Type  KK + <return>  to reset using same file. ";ANS$
1540     IF ANS$ = "KK" THEN 1440
1560     IF ANS$ = "kk" THEN 1440
1580 CLS : SOUND 760,6 : PRINT : INPUT "Type  yes  to run program again. ";ANS$
1600     IF LEFT$(ANS$,1) = "y" THEN 100
1620     IF LEFT$(ANS$,1) = "Y" THEN 100
1640    RUN "MENU.BAS"
1660 END
1680 REM **************** No existing file error trap *************************
1700      IF ERR = 53 THEN PRINT "File not found; try again. " : RESUME 1120
1720      ON ERROR GOTO 0
1740 REM ********************** INTERRUPT SUBROUTINE ***************************
1760        BEEP : BEEP
1780        CLS : LOCATE 4, 1
1800        PRINT : PRINT "PROGRAM INTERRUPT... "
1820        PRINT : PRINT "Type  <RETURN>  to resume this program section."
1840        PRINT : PRINT "Type  <KK>  to start this program section over."
1860        PRINT : PRINT "Type any other key + <RETURN> to exit to main menu."
1880        PRINT : INPUT ANS$
1900          IF ANS$ = "" THEN RETURN
1920          IF ANS$ = "KK" THEN 100
1940          IF ANS$ = "kk" THEN 100
1960          RUN "MENU.BAS"
```

```
100 REM ******************** GPIB-PC PROGRAM HEADER **************************
120        CLEAR   ,59504!:REM   BASIC Declarations
140        IBINIT1 = 59504!
160        IBINIT2 = IBINIT1 + 3:REM Lines 120 through 220 MUST be included
                 in your program.
180        BLOAD "bib.m",IBINIT1
200        CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,
           IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,
           IBWRTF)
220        CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,
           IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,IBWRTI,IBRDIA,
           IBWRTIA,IBSTA%,IBERR%,IBCNT%)
240 REM *************** END OF GPIB-PC PROGRAM HEADER ************************
260 DIM A%(10000)
280   KEY 9, "EXIT" : ON KEY (9) GOSUB 1420 : KEY (9) ON :REM F9 key interrupt.
300 CLS : PRINT : PRINT : REM Program Name <FPLOT.BAS>
320    ON ERROR GOTO 1360 : REM No existing file trap.
340 PRINT "******************  PROGRAM   <FPLOT.BAS>  ************************"
360 PRINT "** This program files signal analyzer screen images in disc     **"
380 PRINT "** files named by the operator.  Each analyzer screen image is  **"
400 PRINT "** stored as an integer array.   Each element of the integer    **"
420 PRINT "** array represents two ASCII code characters from the HP       **"
440 PRINT "** plotter language. (Ref: HP Plotter Prog. Manual, pg. 1-8 & 9)**"
460 PRINT "** Please ensure that the signal analyzer is turned on and      **"
480 PRINT "** that it's GPIB address is twenty-five (25).                  **"
500 PRINT "******************************************************************"
520 PRINT "** NOTE: Use <MCNFG.BAS> to store Signal Analyzer machine       **"
540 PRINT "** configuration files identified as   ''filename.cfg''         **"
560 PRINT "******************************************************************"
580           SOUND 2100, 10 : SOUND 1970, 9
600 PRINT : PRINT "What filename do you wish to specify? "
620 PRINT "Note: File designations must be  dta  (i.e.); filename.dta "
640 PRINT : INPUT "Type your filename.dta";FILE$
660    IF LEN(FILE$) > 12 THEN 580 :REM Limit filename length to 12 characters.
680    IF RIGHT$(FILE$,3) = "DTA" THEN 720
700    IF RIGHT$(FILE$,3) <> "dta" THEN 580
720 V% = 0
740 DEV$ = "DEV1"
760 CNT% = 11000
780 CMD1$ = "PLOT? "
800 CALL IBFIND (DEV$,DV%)
820 CALL IBTMO (DV%, V%)
840 CALL IBWRT (DV%, CMD1$)
860 CALL IBRDI (DV%, A%(0), CNT%)
880 PRINT:PRINT "The following matrices of integers is a sample of how "
    : PRINT "  the analyzer data looks stored on the disc... "
900      FOR N = 1 TO 2400: NEXT N : PRINT : REM This is a time delay line.
920 PRINT "SAMPLE DATA... SAMPLE DATA... SAMPLE DATA... "
940 PRINT "SAMPLE DATA... SAMPLE DATA... SAMPLE DATA... "
960      FOR I = 1 TO 220: PRINT A%(I); :  NEXT I
980 PRINT : PRINT : PRINT "Storing screen data to disc... "
1000 OPEN "C:\GPIB-PC\" + FILE$ FOR OUTPUT AS #1
1020 FOR J = 1 TO 10000 STEP 4
1040    PRINT#1, USING "########"; A%(J); A%(J+1); A%(J+2); A%(J+3);
1060    IF A%(J) + A%(J+1) + A%(J+2) + A%(J+3) = 0 THEN 1120
```

```
1080    PRINT "+";
1100 NEXT J
1120 CLOSE #1
1140 PRINT : PRINT "Screen data stored on disc under filename ";FILE$
1160 PRINT : PRINT FILE$;" disc file filled with ";J;"elements... "
1180 PRINT : PRINT "Disc file ";FILE$;" closed ... "
1200    ERASE A% : REM Erase prior contents of A%(matrix) to prep for reuse.
1220 PRINT:PRINT "Type  yes  to run the program again to file another signal"
1240 INPUT " analyzer screen";ANS$
1260 IF LEFT$(ANS$,1) = "y" THEN 100
1280 IF LEFT$(ANS$,1) = "Y" THEN 100
1300    RUN "MENU.BAS"
1320 END
1340 REM ********** No file/Bad filename error trap routine *******************
1360     IF ERR = 64 THEN PRINT "Bad file name; try again. "
1380     IF ERR = 50 THEN PRINT "Filename is too long; try again. "
          : RESUME 340
1400     ON ERROR GOTO 0
1420 REM ************* SUBROUTINE FOR F9 MENU ESCAPE SEQUENCE *****************
1440      BEEP : BEEP
1460      CLS : LOCATE 4, 1
1480      PRINT "PROGRAM INTERRUPT... "
1500      PRINT : PRINT "Type  <RETURN> to resume this program section."
1520      PRINT : PRINT "Type  <KK>  to start this program section over again."
1540      PRINT : PRINT "Type any other key + <RETURN> to exit to main menu."
1560      PRINT : INPUT ANS$
1580         IF ANS$ = "" THEN RETURN
1600         IF ANS$ = "KK" THEN 100
1620         IF ANS$ = "kk" THEN 100
1640         RUN "menu.bas"
```

```
100 REM **************** Program Name <PRPLOT.BAS> **************************
120 CLEAR : ON ERROR GOTO 1100 : REM Zero variables & set no existing file
    trap.
140   DIM A%(10000) : CLS : REM Dimension A%(matrix) & clear screen.
160   KEY 9, "EXIT" : ON KEY (9) GOSUB 1140 : KEY (9) ON : REM Interrupt trap.
180 REM This program outputs the contents of the disc file designated by the
200 REM   operator to the line printer in the format in which it is stored.
220 PRINT "**************** Program Name <PRPLOT.BAS> **********************"
240 PRINT "** This program prints the contents of a GPIB disc file on the  **"
260 PRINT "** line printer.  The integer disc file is printed in the same  **"
280 PRINT "** format in which it was stored.  Any integer disc file        **"
300 PRINT "** containing less than 10000 integer elements can be printed.  **"
320 PRINT "******************************************************************"
340   PRINT : INPUT "Type <RETURN> to continue... ", ANS$
360   CLS : PRINT "Standard files on this disc are... "
380  PRINT : FILES "*.dta" : FILES "*.CFG" : FILES "*.xyc"
400         SOUND 2100, 10 : SOUND 1970, 9.600001
420 PRINT : PRINT "What filename? "
440 PRINT "Please include a file designation  (i.e.); filename.xxx  "
460 PRINT "  where  xxx  is the designation. "
480 PRINT "[---- NON-INTEGER files will not correctly load. ----]"
500 PRINT : INPUT "Filename.xxx";FILE$
520    IF RIGHT$(FILE$,3) = "bas" THEN 220
540    IF RIGHT$(FILE$,3) = "BAS" THEN 220
560    IF RIGHT$(FILE$,3) = "EXE" THEN 220
580    IF RIGHT$(FILE$,3) = "exe" THEN 220
600    IF RIGHT$(FILE$,3) = "COM" THEN 220
620    IF RIGHT$(FILE$,3) = "com" THEN 220
640    IF RIGHT$(FILE$,3) = "BAT" THEN 220
660    IF RIGHT$(FILE$,3) = "bat" THEN 220
680 PRINT : PRINT "Reading the contents of disc file ";FILE$ : PRINT
700 OPEN "c:\gpib-pc\" + FILE$ FOR INPUT AS #1
720   FOR I = 1 TO 10000 STEP 2
740     IF EOF(1) THEN 840
760       INPUT#1, A%(I), A%(I+1)
780       IF A%(I) + A%(I+1) = 0 THEN 840
800          PRINT "+";
820   NEXT I
840 CLOSE #1
860 PRINT : PRINT : PRINT "Sending file ";FILE$;" to the line printer. "
880 FOR J = 1 TO 10000
900      PRINT USING "########";A%(J);
920   LPRINT USING "########";A%(J);
940   IF A%(J) + A%(J+1) + A%(J+2) = 0 THEN 980
960 NEXT J
980 PRINT : INPUT "Type  yes  to print another file"; ANS$
1000    IF LEFT$(ANS$,1) = "y" THEN 100
1020    IF LEFT$(ANS$,1) = "Y" THEN 100
1040    RUN "MENU.BAS"
1060 END
1080 REM **************** NO EXISTING FILE ERROR TRAP ************************
1100      IF ERR = 53 THEN PRINT "File not found; try again. " : RESUME 220
1120      ON ERROR GOTO 0
1140 REM ************* SUBROUTINE FOR F9 KEY PROGRAM INTERRUPT ****************
1160      CLS : LOCATE 4, 1 : BEEP : BEEP
```

```
1180      PRINT : PRINT "PROGRAM INTERRUPT... "
1200      PRINT : PRINT "Type  <RETURN>  to resume this program section."
1220      PRINT : PRINT "Type  <KK>  to start this program section over."
1240      PRINT : PRINT "Type any other key + <RETURN> to exit to main menu."
1260      PRINT : INPUT ANS$
1280        IF ANS$ = "" THEN RETURN
1300        IF ANS$ = "KK" THEN 100
1320        IF ANS$ = "kk" THEN 100
1340           RUN "menu.bas"
```

```
100 REM ******************* interface board heading ************************
120        CLEAR    ,59504!:REM   BASIC Declarations
140        IBINIT1 = 59504!
160        IBINIT2 = IBINIT1 + 3:REM Lines 120 through 220 MUST be included
                      in your program.
180        BLOAD "bib.m",IBINIT1
200        CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,
           IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,
           IBWRTF)
220        CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,
           IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,IBWRTI,IBRDIA,
           IBWRTIA,IBSTA%,IBERR%,IBCNT%)
240 REM ******************* end of IB program heading *********************
260 REM ---- program name Computer Plotter <CPLTR.BAS> ----
280 DIM A%(10000)
300   KEY 9, "EXIT" : ON KEY (9) GOSUB 1500 : KEY (9) ON : REM Interrupt trap.
320 CLS : SOUND 800, 10: SOUND 780, 8
340 PRINT "************* Program Computer Plotter <CPLTR.BAS>  ***************"
360 PRINT "** This program uses the HP-plotter to draw a graph using data   **"
380 PRINT "** from a computer disc file selected by the operator.  The file **"
400 PRINT "** chosen must contain an image from the Scientific Atlanta      **"
420 PRINT "** signal analyzer that was saved to disc in integer format.     **"
440 PRINT "** The file designation must be (.dta), (i.e.) filename.dta      **"
460 PRINT "** Please ensure that the plotter is connected and turned on.    **"
480 PRINT "*******************************************************************"
500    ON ERROR GOTO 1460 : REM No existing file trap.
520    PRINT : INPUT "Type <RETURN> to continue... ", ANS$
540    CLS
560 PRINT : FILES "*.dta"
580 PRINT : INPUT "What file name contains your graph";FILE$
600    IF RIGHT$(FILE$,3) = "dta" THEN 640
620    IF RIGHT$(FILE$,3) <> "DTA" THEN 320
640 CLS : LOCATE 4, 1
660 PRINT "Loading disc file ";FILE$;" ... Please wait... "
680 CNT% = 10500
700 OPEN "c:\gpib-pc\" + FILE$ FOR INPUT AS #1
720 FOR I = 1 TO 10000 STEP 2
740       IF EOF(1) THEN 880
760   INPUT#1, A%(I), A%(I+1)
780   IF A%(I) + A%(I+1) = 0 THEN 880
800   PRINT "+";
820   X = CSRLIN : IF X <= 22 THEN 860
840     CLS : LOCATE 4, 1 : PRINT "Still loading ";FILE$;
860 NEXT I
880 CLOSE #1
900 REM Program section to initiate HP-plotter and select pen.
920   PRINT : PRINT : PRINT "Plotting the disc file ";FILE$;
940   CMD1$ = ";;IN;;"
960   CMD2$ = ";;SP;;SP;;SP 1;;"
980   CMD3$ = ";;PA1100,2100;;"
1000   CMD4$ = ";;PU;;PU;;SP;;"
1020   DEV$ = "hppltr"
1040   CALL IBFIND (DEV$, DV%)
1060   CALL IBCMD (DV%, CMD1$)
1080   CALL IBWRT (DV%, CMD2$)
```

```
1100   CALL IBWRT (DV%, CMD3$)
1120 REM program section to send data to plotter
1140    V% = 0
1160    CALL IBTMO (DV%, V%)
1180    CALL IBWRTI (DV%, A%(0), CNT%)
1200    CALL IBWRT (DV%, CMD4$)
1220    PRINT : PRINT : PRINT "data bytes written to plotter... "; IBCNT%
1240    PRINT "plotter status... "; IBSTA%
1260 REM program section to end or re-run the program
1280 PRINT : PRINT : PRINT "Type  yes  to run this program again. "
1300 INPUT "Type  K  to print the same disc file again. "; ANS$
1320   IF LEFT$(ANS$,1) = "y" THEN 100
1340   IF LEFT$(ANS$,1) = "Y" THEN 100
1360   IF ANS$ = "K" THEN 920
1380   IF ANS$ = "k" THEN 920
1400      RUN "MENU.BAS"
1420 END
1440 REM *************** No existing file trap ********************************
1460    IF ERR = 53 THEN PRINT "File not found; try again. " : RESUME 340
1480    ON ERROR GOTO 0
1500 REM ****************** KEY (9) INTERRUPT SUBROUTINE ********************
1520       CLS : LOCATE 4, 1 : BEEP : BEEP
1540       PRINT : PRINT "PROGRAM INTERRUPT... "
1560       PRINT : PRINT "Type  <RETURN>  to resume this program section."
1580       PRINT : PRINT "Type  <KK>  to start this program section over."
1600       PRINT : PRINT "Type any other key + <RETURN> to exit to main menu."
1620       PRINT : INPUT ANS$
1640           IF ANS$ = "" THEN RETURN
1660           IF ANS$ = "KK" THEN 100
1680           IF ANS$ = "kk" THEN 100
1700              RUN "menu.bas"
```

```
100 REM **************** Program Name <INTDTA.BAS> ***************************
120   CLEAR : ON ERROR GOTO 1580 : REM Clear screen & set no file trap.
140   DIM A%(10000) : REM Dimension A%(matrix).
160 REM ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
180   KEY 9, "EXIT" : ON KEY (9) GOSUB 1620 : KEY (9) ON : REM Interrupt trap.
200 REM ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
220 REM This program opens a GPIB disc file, reads the integer data into
240 REM   memory, then converts that integer data to ASCII characters for
260 REM   interpretation.  The converted code is displayed on the screen
280 REM   and is also sent as output to the line printer.
300 CLS : SOUND 2100, 10 : SOUND 1970, 9.600001
320 PRINT"***************** Program Name <INTDTA.BAS> **********************"
340 PRINT"** This program converts the integer contents of a GPIB disc file **"
360 PRINT"** into ASCII characters for interpretation.  The converted code  **"
380 PRINT"** is displayed on the screen and output to the line printer.     **"
400 PRINT"** (Non-integer files will not load.) {Interpretation of output:  **"
420 PRINT"** Plotter language is explained in the HP Plotter Prog. Manual   **"
440 PRINT"** & Signal Analyzer codes in the Sci. Atlanta Operating Manual.} **"
460 PRINT"********************************************************************"
480 PRINT"** NOTE: Type F9 to interrupt program printing operation if the   **"
500 PRINT"** entire decoded file is not desired from the line printer.      **"
520 PRINT"********************************************************************"
540    PRINT : INPUT "Type <RETURN> to continue... ", ANS$
560    CLS
580    FILES
600 PRINT : PRINT "What filename? "
620 PRINT "Please include a file designation  (i.e.); filename.xxx  "
640 PRINT " where  xxx  is the designation (BAS, BAT, COM or EXE are invalid)."
660 PRINT : INPUT "Filename.xxx";FILE$
680 IF RIGHT$(FILE$,3) = "bas" THEN 300
700 IF RIGHT$(FILE$,3) = "BAS" THEN 300
720 IF RIGHT$(FILE$,3) = "EXE" THEN 300
740 IF RIGHT$(FILE$,3) = "exe" THEN 300
760 IF RIGHT$(FILE$,3) = "COM" THEN 300
780 IF RIGHT$(FILE$,3) = "com" THEN 300
800 IF RIGHT$(FILE$,3) = "BAT" THEN 300
820 IF RIGHT$(FILE$,3) = "bat" THEN 300
840 PRINT : PRINT "Reading the contents of disc file ";FILE$ : PRINT
860 OPEN "c:\gpib-pc\" + FILE$ FOR INPUT AS #1
880 FOR I = 1 TO 10000 STEP 4
900   IF EOF(1) THEN 1000 : REM End of File (EOF) trap.
920   INPUT#1, A%(I), A%(I+1), A%(I+2), A%(I+3)
940   IF A%(I) + A%(I+1) + A%(I+2) + A%(I+3) = 0 THEN 1000
960     PRINT "+";
980 NEXT I
1000 CLOSE #1
1020 PRINT : PRINT : PRINT "The following is sample data from disc file ";FILE$
1040 PRINT "-----------------------------------------------------------------"
1060 PRINT "SAMPLE DATA... SAMPLE DATA... SAMPLE DATA... "
1080 PRINT "SAMPLE DATA... SAMPLE DATA... SAMPLE DATA... "
1100 PRINT "-----------------------------------------------------------------"
1120 FOR I = 1 TO 200: PRINT A%(I); :  NEXT I
1140 PRINT
1160 PRINT : PRINT "ASCII interpretations.............................. "
1180 LPRINT : LPRINT "ASCII interpretations.............................. "
```

```
1200 FOR K = 1 TO 10000 STEP 4
1220 BB = INT(A%(K)/256) : AA = A%(K) - (BB*256)
1240 BB1 = INT(A%(K+1)/256) : AA1 = A%(K+1) - (BB1*256)
1260 BB2 = INT(A%(K+2)/256) : AA2 = A%(K+2) - (BB2*256)
1280 BB3 = INT(A%(K+3)/256) : AA3 = A%(K+3) - (BB3*256)
1300 PRINT A%(K);SPC(1);A%(K+1);SPC(1);A%(K+2);SPC(1);A%(K+3);SPC(2);
1320 LPRINT A%(K);SPC(1);A%(K+1);SPC(1);A%(K+2);SPC(1);A%(K+3);SPC(2);
1340 PRINT CHR$(AA);SPC(1);CHR$(BB);SPC(1);CHR$(AA1);SPC(1);CHR$(BB1);SPC(1);
1360 LPRINT CHR$(AA);SPC(1);CHR$(BB);SPC(1);CHR$(AA1);SPC(1);CHR$(BB1);SPC(1);
1380 PRINT CHR$(AA2);SPC(1);CHR$(BB2);SPC(1);CHR$(AA3);SPC(1);CHR$(BB3)
1400 LPRINT CHR$(AA2);SPC(1);CHR$(BB2);SPC(1);CHR$(AA3);SPC(1);CHR$(BB3)
1420 IF (A%(K) + A%(K+1) + A%(K+2) + A%(K+3) = 0) THEN 1460
1440 NEXT K
1460 PRINT : INPUT "Type  yes  to print another file"; ANS$
1480 IF LEFT$(ANS$,1) = "y" THEN 100
1500 IF LEFT$(ANS$,1) = "Y" THEN 100
1520   RUN "MENU.BAS"
1540 END
1560 REM ********************** No existing file error trap ********************
1580     IF ERR = 53 THEN PRINT "File not found; try again. "
          : SOUND 600, 6 : SOUND 580, 4 : RESUME 320
1600     ON ERROR GOTO 0
1620 REM ****************** KEY (9) INTERRUPT SUBROUTINE *********************
1640       CLS : LOCATE 4, 1 : BEEP : BEEP
1660       PRINT : PRINT "PROGRAM INTERRUPT... "
1680       PRINT : PRINT "Type  <RETURN>  to resume this program section."
1700       PRINT : PRINT "Type  <KK>  to start this program section over."
1720       PRINT : PRINT "Type any other key + <RETURN> to exit to main menu."
1740       PRINT : INPUT ANS$
1760          IF ANS$ = "" THEN RETURN
1780          IF ANS$ = "KK" THEN 100
1800          IF ANS$ = "kk" THEN 100
1820             RUN "menu.bas"
```

```
100 REM ******************** GPIB Program Header ******************************
120       CLEAR   ,59504!:REM   BASIC Declarations
140       IBINIT1 = 59504!
160       IBINIT2 = IBINIT1 + 3:REM Lines 120 through 220 MUST be included
                   in your program.
180       BLOAD "bib.m",IBINIT1
200       CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,
          IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,
          IBWRTF)
220       CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,
          IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,IBWRTI,IBRDIA,
          IBWRTIA,IBSTA%,IBERR%,IBCNT%)
240 REM ************************************************************************
260 REM +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
280   KEY 9, "EXIT" : ON KEY (9) GOSUB 2580 : KEY (9) ON : REM Interrupt trap.
300 REM +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
320 REM ----------  Program Graph Data <GRDTA.BAS>  ------------
340 DIM A%(10100), B%(6000)
360 REM A%(xxxxx) is a matrix used to store signal analyzer graphic data from
         the designated disc file into the computer active memory
380 REM B%(xxxxx) is a matrix used to store the selected graphic portion of
         the signal analyzer disc file to be plotted via the HP plotter
400 CLS : REM Fol box appears on the screen and defines the program function
420 SOUND 940, 10 : SOUND 860, 9
440 PRINT"**************** Program <GRDTA.BAS> ******************************"
460 PRINT"** This program loads the contents of a designated GPIB disc file **"
480 PRINT"** containing integer data into the computer; identifies the file **"
500 PRINT"** graphic data section, then plots that disc file graphic data   **"
520 PRINT"** using the Hewlett Packard plotter.  This program can be used   **"
540 PRINT"** to plot several signal analyzer graphic displays on the same   **"
560 PRINT"** page, providing direct comparison of different sample results. **"
580 PRINT"*******************************************************************"
600 PRINT"** NOTE: Curves plotted together on the same plotter display      **"
620 PRINT"** should all have originated from equivalent coordinate scales   **"
640 PRINT"** when they initially were displayed on the Signal Analyzer.     **"
660 PRINT"*******************************************************************"
680        ON ERROR GOTO 2540 : REM No existing file error trap.
700    PRINT : INPUT "Type <RETURN> to continue... ", ANS$
720    CLS : REM Clear screen.
740 PRINT : FILES "*.dta"
760 PRINT : PRINT "What file name contains your data?"
780 PRINT "File designation must be  dta  (ie); filename.dta  "
800 PRINT "Note: File desig's <EXE>, <BAS>, <BAT> & <COM> will not load."
820 PRINT : INPUT "Filename.dta ...";FILE$
840    IF RIGHT$(FILE$,3) = "DTA" THEN 1000
860    IF RIGHT$(FILE$,3) = "Dta" THEN 1000
880    IF RIGHT$(FILE$,3) = "dta" THEN 1000
900    IF RIGHT$(FILE$,3) = "dTA" THEN 1000
920    IF RIGHT$(FILE$,3) = "dtA" THEN 1000
940    IF RIGHT$(FILE$,3) = "dTa" THEN 1000
960    IF RIGHT$(FILE$,3) = "DTa" THEN 1000
980       GOTO 400
1000 PRINT:PRINT "Loading the contents of "; FILE$; " into the computer... "
1020 PRINT:PRINT "Please wait... "
1040 OPEN "c:\gpib-pc\" + FILE$ FOR INPUT AS #1
```

```
1060 FOR I = 1 TO 10000 STEP 4
1080   IF EOF(1) THEN 1240 : REM  End of File test.
1100   INPUT#1, A%(I), A%(I+1), A%(I+2), A%(I+3)
1120   IF A%(I) + A%(I+1) + A%(I+2) + A%(I+3) = 0 THEN 1240
1140   PRINT "+";
1160   X = CSRLIN : IF X <= 23 THEN 1200
1180       CLS: LOCATE 4,1 : PRINT "Still loading ";FILE$;
1200 NEXT I
1220 REM FE = the number of the last graphic data byte (or ASCII integer)
1240 FE = I : CLOSE #1
1260 CLS: PRINT : PRINT "Disc file ";FILE$; " is loaded into active memory... "
1280 PRINT : PRINT "Identifying the start of graphic data ..."
1300 REM The following program section searches through the integer disc file
        and locates the graphic signal analyzer data.
1320 REM The graphic data begins with either (;; PD = 15163 17488) or
        (;P D; = 20539 15172) or (PD ;; = 17488 15163) or (D;;P = 15172 20539).
1340 REM Therefore, we only need check for the integers (17488 = PD) or for
        (15172 = D;) which are common to all four possibilities.
1360 REM The graphic data ends with either (;; PU = 15163 21840) or
        (;P U; = 20539 15189) or (PU ;; = 21840 15163) or (U;;L = 15189 19515).
1380 REM Therefore, we only need check for the integers (21840 = PU) or for
        (15189 = U;) which are common to all four possibilities.
1400     N = -100
1420   FOR J = 1 TO 10000
1440      IF N > 35 THEN 1580
1460      IF A%(J) = 17488 THEN 1500
1480      IF A%(J) <> 15172 THEN 1540
1500         SB% = J : IF N > 35 THEN 1580
1520         N = 0
1540      N = N + 1 : PRINT "!";
1560 NEXT J
1580 REM ** THE STARTING INTEGER FOR GRAPHIC DATA IS NOW SET AS  "SB%" **
1600 PRINT : CLS : PRINT "Locating the end of graphic data ... "
1620   FOR K = SB% + 50 TO FE
1640      IF A%(K) = 21840 THEN 1680
1660      IF A%(K) <> 15189 THEN 1700
1680         EB% = K : IF EB% - SB% > 100 THEN 1740
1700      PRINT "#";
1720   NEXT K
1740 REM ** THE ENDING INTEGER FOR GRAPHIC DATA IS NOW SET AS  "EB%" **
1760 REM Place the identified graphic data from A%(matrix) into B%(matrix).
1780 PRINT : CLS : PRINT "Storing the identified graphic data in B%(matrix)."
1800 REM The first ten integers of the file are graphic scaling instructions.
1820   FOR KK = 1 TO 10 : B%(KK) = A%(KK) : NEXT KK
1840 REM The following three integers institute a "pen down" instruction.
1860   B%(11) = 15163 : B%(12) = 17488 : B%(13) = 15163
1880 REM The following loop places the actual graphic data into B%(matrix).
1900   FOR L = 14 TO (EB% - SB% +1)
1920     B%(L) = A%(L + SB% - 14)
1940     PRINT "@";
1960   NEXT L
1980 REM The fol program section plots the graphic data previously identified.
2000   PRINT : CLS : PRINT "Plotting the selected graphic data ... "
2020   CMD1$ = ";;IN;;"
2040   CMD2$ = ";;SP;;SP;;SP1;;;PA1100,2100;;;"
```

```
2060   CMD3$ = ";;PU;;PU;;SP;;"
2080   V% = 0 : CNT% = 9000
2100   DEV2$ = "hppltr"
2120   CALL IBFIND (DEV2$, DV%)
2140   CALL IBCMD (DV%, CMD1$)
2160   CALL IBWRT (DV%, CMD2$)
2180   CALL IBTMO (DV%, V%)
2200       PRINT : PRINT "Calling IBWRTI( B%(matrix) )... "
2220       PRINT "The interface board function to plot the data... "
2240   CALL IBWRTI (DV%, B%(0), CNT%)
2260   CALL IBWRT (DV%, CMD3$)
2280 PRINT : PRINT "Starting integer # = ";SB% ;"  Ending integer # = ";EB%
2300 PRINT : PRINT "Type  <KK>  to RE-PLOT the same graph."
2320         PRINT "(Reposition graph paper to origin for a re-plot.)"
2340 PRINT : INPUT "Type  yes  to run the program again..."; ANS$
2360    IF ANS$ = "KK" THEN 1980
2380    IF ANS$ = "kk" THEN 1980
2400    IF ANS$ = "Kk" THEN 1980
2420    IF ANS$ = "kK" THEN 1980
2440    IF LEFT$(ANS$,1) = "y" THEN 100
2460    IF LEFT$(ANS$,1) = "Y" THEN 100
2480       RUN "MENU.BAS"
2500 END
2520 REM ********************* No existing file error trap *********************
2540      IF ERR = 53 THEN PRINT "File not found; try again." : RESUME 420
2560      ON ERROR GOTO 0
2580 REM ****************** KEY (9) INTERRUPT SUBROUTINE *********************
2600       CLS : LOCATE 4, 1 : BEEP : BEEP
2620       PRINT : PRINT "PROGRAM INTERRUPT... "
2640       PRINT : PRINT "Type  <RETURN>  to resume this program section."
2660       PRINT : PRINT "Type  <KK>  to start this program section over."
2680       PRINT : PRINT "Type any other key + <RETURN> to exit to main menu."
2700       PRINT : INPUT ANS$
2720          IF ANS$ = "" THEN RETURN
2740          IF ANS$ = "KK" THEN 100
2760          IF ANS$ = "kk" THEN 100
2780             RUN "menu.bas"
```

```
100 REM --------------  Program Screen Data <SCRNDTA.BAS>  ----------------
120       CLEAR : REM Zero all variables.
140       DIM A%(10000) : REM Dimension A%(matrix).
160 REM A%(xxxxx) is a matrix used to store signal analyzer graphic data from
    the designated disc file into the computer active memory
180 CLS : REM Fol box appears on the screen and defines the program function
200 SOUND 2010, 10 : SOUND 1980, 9
220 PRINT"************ Program to display disc file <SCRNDTA.BAS> ************"
240 PRINT"** This program loads the contents of a designated GPIB disc file **"
260 PRINT"** containing integer data into the computer and then displays    **"
280 PRINT"** the contents and its decoded ASCII characters for analysis.    **"
300 PRINT"** Note: File desig's can not be <BAT>, <BAS>, <EXE> or <COM>.    **"
320 PRINT"*******************************************************************"
340 REM ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
360   KEY 9, "EXIT" : ON KEY (9) GOSUB 1780 : KEY (9) ON : REM Interrupt trap.
380 REM ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
400       ON ERROR GOTO 1680 : REM Set no existing file trap.
420    PRINT : INPUT "Type <RETURN> to continue... ", ANS$
440    CLS : REM Clear screen.
460 PRINT : FILES
480 PRINT : INPUT "What file name contains your data";FILE$
500    IF RIGHT$(FILE$,3) = "exe" THEN 180
520    IF RIGHT$(FILE$,3) = "EXE" THEN 180
540    IF RIGHT$(FILE$,3) = "bas" THEN 180
560    IF RIGHT$(FILE$,3) = "BAS" THEN 180
580    IF RIGHT$(FILE$,3) = "bat" THEN 180
600    IF RIGHT$(FILE$,3) = "BAT" THEN 180
620    IF RIGHT$(FILE$,3) = "com" THEN 180
640    IF RIGHT$(FILE$,3) = "COM" THEN 180
660 PRINT:PRINT "Loading the contents of "; FILE$; " into the computer... "
680 PRINT:PRINT "Please wait... "
700 OPEN "c:\gpib-pc\" + FILE$ FOR INPUT AS #1
720   FOR I = 1 TO 10000 STEP 4
740       IF EOF(1) THEN 900
760     INPUT#1, A%(I), A%(I+1), A%(I+2), A%(I+3)
780     IF A%(I) + A%(I+1) + A%(I+2) + A%(I+3) = 0 THEN 900
800        PRINT "+";
820        X = CSRLIN : IF X <= 22 THEN 860
840        CLS: LOCATE 4,1 : PRINT "Still loading ";FILE$;
860   NEXT I
880 REM FE = variable to hold the number of the last graphic data byte
900   FE = I : CLOSE #1
920   ON ERROR GOTO 1740 : REM Set trap for integer overflow.
940 CLS: PRINT "Disc file ";FILE$; " is loaded into active memory ... "
960 REM SB% = starting byte of desired data section   EB% = ending byte of
        desired data section
980 SOUND 1900, 8 : SOUND 1880, 6
1000 PRINT:PRINT "Designate the number of the starting byte... ";:INPUT SB%
1020 IF SB% < 1 THEN 980 : REM this line traps invalid byte designations that
        are less than the beginning of the graphic data file.
1040 IF SB% > FE THEN 980 : REM this line traps starting byte designations
        that are greater than the end of the data file.
1060 SOUND 1900, 8 : SOUND 1880, 6
1080 PRINT:PRINT "Designate the number of the ending byte... ";:INPUT EB%
1100 IF EB% > FE THEN 1060 : REM this line traps invalid byte designations
```

```
       that are greater than the end of the data file.
1120 IF EB% < SB% THEN 1060 : REM this line traps invalid ending byte
              designations that are less than the starting byte value.
1140 CLS: PRINT "Your chosen data bytes contain... " : PRINT
1160 REM following line prints output data column headings
1180 PRINT "Byte #  Byte #  Integer  Integer  ASCII Characters "
1200    FOR I = SB% TO EB%  STEP 2
1220      PRINT SPC(2);I;SPC(2);I+1;SPC(2);
1240      PRINT A%(I);SPC(1);A%(I + 1);SPC(3);
1260      REM the following lines calculate the ASCII Codes equivalent to the
              integers stored in matrix A%(xxxxx)
1280      BB = INT(A%(I)/256) : AA = A%(I) - (BB*256)
1300      BB1 = INT(A%(I + 1)/256) : AA1 = A%(I + 1) - (BB1*256)
1320      PRINT CHR$(AA);SPC(1);CHR$(BB);SPC(1);CHR$(AA1);SPC(1);CHR$(BB1)
1340      REM the fol two lines limit screen output to nineteen lines at a
           time SLINE% = the number of <SCREEN LINES> currently displayed
           as output
1360        SLINE% = SLINE% + 1 : IF SLINE% >= 19 THEN 1380 ELSE 1440
1380        SLINE% = 0 : INPUT "Type return for next screen... ";ANS$ : CLS
1400        REM the fol line prints output data column headings
1420        PRINT "Byte #  Byte #  Integer  Integer  ASCII Characters "
1440    NEXT I
1460 SLINE% = 0 : REM reset screen line number variable
1480 PRINT : PRINT "Type  yes  to look at other data bytes... ";: INPUT ANS$
1500    IF LEFT$(ANS$,1) = "y" THEN 940
1520    IF LEFT$(ANS$,1) = "Y" THEN 940
1540 PRINT : PRINT "Graphic file contained ";FE - 4;" total bytes of data."
1560 PRINT : INPUT "Type  yes  to examine another disc file... "; ANS$
1580    IF LEFT$(ANS$,1) = "y" THEN 100
1600    IF LEFT$(ANS$,1) = "Y" THEN 100
1620        RUN "MENU.BAS"
1640 END
1660 REM ************** No existing file trap routine ************************
1680      IF ERR = 53 THEN PRINT "File not found; try again. " : RESUME 200
1700      ON ERROR GOTO 0
1720 REM ************** Integer overflow trap routine ************************
1740 IF ERR = 6 THEN PRINT "INTEGER OVERFLOW GW BASIC (Table A1); TRY AGAIN."
      :  RESUME 960
1760      ON ERROR GOTO 0
1780 REM ****************** KEY (9) INTERRUPT SUBROUTINE ********************
1800       CLS : LOCATE 4, 1 : BEEP : BEEP
1820       PRINT : PRINT "PROGRAM INTERRUPT... "
1840       PRINT : PRINT "Type  <RETURN>  to resume this program section."
1860       PRINT : PRINT "Type  <KK>  to start this program section over."
1880       PRINT : PRINT "Type any other key + <RETURN> to exit to main menu."
1900       PRINT : INPUT ANS$
1920          IF ANS$ = "" THEN RETURN
1940          IF ANS$ = "KK" THEN 100
1960          IF ANS$ = "kk" THEN 100
1980             RUN "menu.bas"
```

```
100 REM ********************* Program <ITERPLOT.BAS> **************************
120        CLEAR   ,59504!:REM   BASIC Declarations
140        IBINIT1 = 59504!
160        IBINIT2 = IBINIT1 + 3:REM Lines 120 through 220 MUST be included
180        BLOAD "bib.m",IBINIT1
200        CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,
           IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,
           IBWRTF)
220        CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,
           IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,IBWRTI,IBRDIA,
           IBWRTIA,IBSTA%,IBERR%,IBCNT%)
240 REM *************** end of gpib-pc interface board header ****************
260 REM +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
280   KEY 9, "EXIT" : ON KEY (9) GOSUB 2140 : KEY (9) ON : REM Interrupt trap.
300 REM +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
320 DIM A%(10000), B%(10000) : REM Dimension A%(matrix) & B%(matrix).
340      ON ERROR GOTO 2020 : REM Set no existing file error trap.
360      SOUND 900,10 : SOUND 800,8 : CLS : REM Beep & clear screen.
380 PRINT "***************** Program <ITERPLOT.BAS> **********************"
400 PRINT "** Program to selectively plot data from an operator desig'd  **"
420 PRINT "** disc file.  This program runs the HP-plotter to draw a     **"
440 PRINT "** graph using designated portions of graphic data from a     **"
460 PRINT "** disc file that was originally obtained from the Scientific **"
480 PRINT "** Atlanta signal analyzer screen display by <FPLOT.BAS>.     **"
500 PRINT "****************************************************************"
520   PRINT : INPUT "Type <RETURN> to continue... ",ANS$
540    CLS
560 PRINT : FILES "*.dta"
580 PRINT : PRINT "What disc file do you wish to plot? "
600 PRINT "Note: File designation must be  dta  (i.e.) filename.dta "
620 PRINT : INPUT "What filename.dta   ";FILE$
640    IF RIGHT$(FILE$,3) = "dta" THEN 760
660    IF RIGHT$(FILE$,3) = "Dta" THEN 760
680    IF RIGHT$(FILE$,3) = "DTa" THEN 760
700    IF RIGHT$(FILE$,3) = "dtA" THEN 760
720    IF RIGHT$(FILE$,3) = "dTA" THEN 760
740    IF RIGHT$(FILE$,3) <> "DTA" THEN 360
760 PRINT : PRINT "Loading ";FILE$;" ... Please wait... "
780 CNT% = 12000
800 OPEN "c:\gpib-pc\" + FILE$ FOR INPUT AS #1
820   FOR I = 1 TO 10000 STEP 4
840       IF EOF(1) THEN 980
860     INPUT#1, A%(I), A%(I+1), A%(I+2), A%(I+3)
880       IF A%(I) + A%(I+1) + A%(I+2) + A%(I+3) = 0 THEN 980
900     PRINT "+";
920     X = CSRLIN : IF X < 22 THEN 960
940         CLS : LOCATE 4,1 : PRINT "Still loading ";FILE$
960   NEXT I
980 EF% = I : CLOSE #1 : REM Variable EF% is the End of Disc File index.
1000 CLS : PRINT "Note: The first several data bytes from the disc file
     contain graphic " : PRINT "scale and pen positioning instructions.
     Thus, you need at least the first " : PRINT "ten bytes to produce
     a readable graphic segment on the HP plotter."
1020 PRINT : PRINT "Designate the number of bytes you wish to use from the "
     : PRINT " beginning of the disc file (scale, border, etc.) " ;
```

```
      : INPUT BF%
1040    IF BF% > EF% THEN 1020 : REM trap for byte number designation that is
               greater than the length of the file
1060    IF BF% < 0 THEN 1020 : REM trap for byte designation that is
                       less than zero
1080 PRINT : INPUT "Designate the starting byte for the graphic data... "; SB%
1100    IF SB% < BF% THEN 1080 : REM trap for starting byte less than BF%
1120    IF SB% < 0 THEN 1080 : REM trap for starting byte less than zero
1140    IF SB% > EF% THEN 1080 : REM trap for starting byte greater than EF%
1160 PRINT : INPUT "Designate the ending byte for the graphic data... " ; EB%
1180    IF EB% < SB% THEN 1160 : REM trap for ending byte value less than
          starting byte value
1200    IF EB% < 0 THEN 1160 : REM trap for ending byte value less than zero
1220    IF EB% > EF% THEN 1160 : REM trap for ending byte value greater than
          the disc file length
1240 PRINT:PRINT "Rearranging original graphic data as requested... "
1260       FOR J = 1 TO BF% : B%(J) = A%(J) : NEXT J
1280 B%(BF% + 1) = 17488 : B%(BF% + 2) = 11514 : B%(BF% + 3) = 15383
     : B%(BF% + 4) = 15163 : REM initiate pen position to P1
     (HP-prog manual, pg. 2-2)
1300 N = 0 : REM set index for A%(matrix) starting byte
1320 REM Set B%(matrix) equal to chosen graphic data portion of A%(matrix)
1340   FOR K = BF% + 5 TO EB% - (SB% - BF% - 1)
1360       B%(K) = A%(SB% + N)
1380       N = N + 1
1400   NEXT K
1420 REM Program section to address & activate the hp-plotter.
1440   CMD1$ = ";;;ATN;;;"
1460   CMD2$ = ";;IN;;"
1480   CMD3$ = ";;;PA1100,2100;;;"
1500   DEV$ = "HPPLTR"
1520   CALL IBFIND (DEV$, DV%)
1540   CALL IBCMD (DV%, CMD1$)
1560   CALL IBCMD (DV%, CMD2$)
1580   CALL IBWRT (DV%, CMD3$)
1600 REM program section to send selected data to plotter
1620 PRINT : PRINT "Plotting the rearranged data... "
1640    V% = 0 : CMD4$ = ";;;PU;;PU;;SP;;;"
1660    CALL IBTMO (DV%, V%)
1680    CALL IBWRTI (DV%, B%(0), CNT%)
1700    CALL IBWRT (DV%, CMD4$)
1720    PRINT "plotter status... "; IBSTA%
1740 PRINT : PRINT "Type  K + <RETURN>  to plot the same graph. "
1760 PRINT : PRINT "Type  I + <RETURN>  to select and plot different portions"
1780 PRINT "             of the same graph. "
1800 PRINT : PRINT "Type  <YES> + <RETURN> to select another disc file."
1820 PRINT : INPUT "Type  <RETURN>  to exit to main menu.  ",ANS$
1840    IF ANS$ = "K" THEN 1420
1860    IF ANS$ = "I" THEN 2060
1880    IF ANS$ = "k" THEN 1420
1900    IF ANS$ = "i" THEN 2060
1920    IF LEFT$(ANS$,1) = "y" THEN 100
1940    IF LEFT$(ANS$,1) = "Y" THEN 100
1960       RUN "MENU.BAS"
1980 END
```

```
2000 REM ************** No existing file error trap routine ******************
2020      IF ERR = 53 THEN PRINT "File not found; try again. " : RESUME 380
2040      ON ERROR GOTO 0
2060 REM ************** Routine to zero B%(matrix) in prep for reuse **********
2080     ERASE B% : REM Erase B%(matrix) to clear previous chosen elements.
2100     DIM B%(10000) : REM Redimension B%(matrix) to store next set elements.
2120          GOTO 1000
2140 REM ****************** KEY (9) INTERRUPT SUBROUTINE *********************
2160       CLS : LOCATE 4, 1 : BEEP : BEEP
2180       PRINT : PRINT "PROGRAM INTERRUPT... "
2200       PRINT : PRINT "Type  <RETURN>  to resume this program section."
2220       PRINT : PRINT "Type  <KK>  to start this program section over."
2240       PRINT : PRINT "Type any other key + <RETURN> to exit to main menu."
2260       PRINT : INPUT ANS$
2280          IF ANS$ = "" THEN RETURN
2300          IF ANS$ = "KK" THEN 100
2320          IF ANS$ = "kk" THEN 100
2340             RUN "menu.bas"
```

```
100 REM ************************************************************************
120      CLEAR : REM Clear memory for start or restart.
140 REM -----------  Program Graph Data <DAMPCALC.BAS>  -------------
160 DIM A%(10000), B%(4000)
180 REM A%(xxxxx) is a matrix used to store signal analyzer graphic data in
          integer form from the designated disc file
200 REM A%(xxxxx) is RE-USED to store the decoded analyzer XY-coordinates
220 REM B%(xxxxx) is a matrix to store the identified graphic portion of
            the analyzer integer file required for damping calculations
240 REM C$(xxxxx) is a matrix to store the decoded ASCII graphic data
            deciphered from the orig analyzer integer disc file
260 REM ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
280   KEY 9, "EXIT" : ON KEY (9) GOSUB 4920 : KEY (9) ON : REM Interrupt trap.
300 REM ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
320 CLS : REM Fol box appears on the screen and defines the program function
340       SOUND 900, 10 : SOUND 840, 9
360 PRINT"**************** Program <DAMPCALC.BAS> ***************************"
380 PRINT"** This program loads the integer contents of a designated GPIB   **"
400 PRINT"** disc file into the computer; identifies the file graphic data  **"
420 PRINT"** section; then calculates the Specific Damping Capacity (SDC)   **"
440 PRINT"** and the Damping Coefficient (DC) for the selected data file.   **"
460 PRINT"** SDC & DC are calculated in absolute HP-plotter coord terms     **"
480 PRINT"** and may vary some from Signal Analyzer values.  This program   **"
500 PRINT"** will also store the S/A graphic XY-coordinates as absolute     **"
520 PRINT"** HP-plotter integer magnitudes, if desired.   (The program      **"
540 PRINT"** <GRAPHXYC.BAS> can display stored disc file graphic data on    **"
560 PRINT"** the microcomputer screen. (Selection #10 from the Main Menu))  **"
580 PRINT"*******************************************************************"
600 PRINT"** Note: This program will not function correctly with signal     **"
620 PRINT"** analyzer dual screen traces that are stored to disc.  GPIB     **"
640 PRINT"** disc data files must be single traces of amplitude vs freq.    **"
660 PRINT"*******************************************************************"
680    ON ERROR GOTO 4880 : REM No existing disc file error trap.
700    PRINT : INPUT "Type <RETURN> to continue... ", ANS$
720    CLS : REM Clear screen.
740 PRINT : FILES "*.dta"
760 PRINT : PRINT "What file name contains your data?"
780 PRINT "File designation must be  dta  (ie);  filename.dta  "
800 PRINT : INPUT "Filename.dta ...";FILE$
820 IF RIGHT$(FILE$,3) = "DTA" THEN 960
840 IF RIGHT$(FILE$,3) = "DTa" THEN 960
860 IF RIGHT$(FILE$,3) = "Dta" THEN 960
880 IF RIGHT$(FILE$,3) = "dTA" THEN 960
900 IF RIGHT$(FILE$,3) = "dtA" THEN 960
920 IF RIGHT$(FILE$,3) <> "dta" THEN 320
940 REM ***********************************************************************
960 CLS : PRINT "Loading the contents of "; FILE$; " into the computer... "
980 PRINT : PRINT "Disc file contents are ... Please wait... "
1000 OPEN "c:\gpib-pc\" + FILE$ FOR INPUT AS #1
1020    FOR I = 1 TO 10000 STEP 4
1040          IF EOF(1) THEN 1200
1060       INPUT#1, A%(I), A%(I+1), A%(I+2), A%(I+3)
1080       IF A%(I) + A%(I+1) + A%(I+2) = 0 THEN 1200
1100       PRINT A%(I),A%(I+1),A%(I+2),A%(I+3),;
1120         X = CSRLIN : IF X <= 23 THEN 1160
```

```
1140         CLS: LOCATE 4,1 : PRINT "Loading contents of ";FILE$
1160    NEXT I
1180 REM FE = the index <File End> of the last graphic data byte
                                                    (or ASCII integer)
1200         FE = I : CLOSE #1
1220 CLS: PRINT : PRINT "Disc file ";FILE$; " is loaded into active memor ..."
1240 REM ****************  ID GRAPHIC DATA START  ******************************
1260 PRINT : PRINT "Identifying the start of graphic data ..."
1280 REM The following program section searches through the integer disc file
        and locates the graphic signal analyzer data.
1300 REM The graphic data begins with either {;; PD = 15163 17488} or
      {;P D; = 20539 15172} or {PD ;; = 17488 15163} or {D;;P = 15172 20539}.
1320 REM Therefore, we only need check for the integers {17488 = PD} or for
      {15172 = D;} which are common to all four possibilities.
1340 REM The graphic data ends with either {;; PU = 15163 21840} or
      {;P U; = 20539 15189} or {PU ;; = 21840 15163} or {U;;L = 15189 19515}.
1360 REM Therefore, we only need check for the integers {21840 = PU} or for
      {15189 = U;} which are common to all four possibilities.
1380     N = -100 : REM Set N < 0, "N" is a test index for next loop.
1400 REM ******************************************************************
1420 REM Fol loop finds starting index (SB%) for the graphic data.
1440    FOR J = 1 TO 10000
1460      IF A%(J) = 17488 THEN 1500
1480      IF A%(J) <> 15172 THEN 1540
1500         SB% = J
1520         N = 0
1540      IF N > 35 THEN 1600 : REM Identifies a graphic data section > 35
                characters long.
1560         N = N + 1 : PRINT "!";
1580    NEXT J
1600 REM ** THE STARTING INTEGER FOR GRAPHIC DATA IS NOW SET AS  "SB%" **
1620 REM ****************  ID END OF GRAPHIC DATA  *************************
1640 PRINT : CLS : PRINT "Locating the end of graphic data ... "
1660 REM Fol loop finds ending index (EB%) for the graphic data.
1680    FOR K = SB% + 50 TO FE
1700      IF A%(K) = 21840 THEN 1760
1720      IF A%(K) <> 15189 THEN 1780
1740      REM {EB% - SB% > 100} identifies a graphic section > 100
                           characters long.
1760         EB% = K : IF EB% - SB% > 100 THEN 1820
1780      PRINT "#";
1800    NEXT K
1820 REM ** THE ENDING INTEGER FOR GRAPHIC DATA IS NOW SET AS  "EB%" **
1840 PRINT : PRINT "Starting integer # = ";SB% ;"  Ending integer # = ";EB%
1860    FOR KK = 1 TO 2000: NEXT KK : REM A delay line to display SB% & EB%.
1880 REM ***************  STORE GRAPHIC DATA IN B%(matrix)  ****************
1900 REM Place the identified graphic data from A%(matrix) into B%(matrix).
1920 PRINT : CLS : PRINT "Storing the identified integer data in B%(matrix)."
1940 REM The following loop places the actual graphic data into B%(matrix).
1960     FOR L = 1 TO (EB% - SB% +1)
1980       B%(L) = A%(L + SB% - 1)
2000       PRINT B%(L);SPC(1);
2020     NEXT L
2040 REM *************  ZERO A%(matrix) & DIM C$(matrix)  *********************
2060       ERASE A% : REM Erasing A%(matrix) to conserve memory space.
```

```
2080       DIM C$(8000) : REM Dimension C$(matrix) for ASCII data.
2100 REM -----------------------------------------------------------------
2120 REM ********* Program section to decode & display graphic data **********
2140 PRINT : PRINT : PRINT "Decoding the integer graphic data."
2160 PRINT : PRINT "Storing the decoded ASCII graphic data in C$(matrix)."
2180   FOR KK = 1 TO 1800 : NEXT KK : REM Delay line for screen display.
2200 PRINT : PRINT "Your chosen data bytes contain... " : PRINT
2220 REM following line prints screen data column headings
2240 PRINT " Byte #  Byte #  Integer  Integer  ASCII Characters "
2260    FOR I = 1 TO EB% - SB%  STEP 2
2280      PRINT SPC(2);I;SPC(2);I+1;SPC(2);
2300      PRINT B%(I);SPC(1);B%(I + 1);SPC(3);
2320      REM the following lines calculate the ASCII Codes equivalent to the
             integers stored in matrix B%(xxxxx)
2340      BB = INT(B%(I)/256) : AA = B%(I) - (BB*256)
2360      BB1 = INT(B%(I + 1)/256) : AA1 = B%(I + 1) - (BB1*256)
2380      PRINT CHR$(AA);SPC(1);CHR$(BB);SPC(1);CHR$(AA1);SPC(1);CHR$(BB1)
2400      REM Fol two lines store decoded graphic data in C$(matrix).
2420        C$(2*I - 1) = CHR$(AA) : C$(2*I) = CHR$(BB)
2440        C$(2*I + 1) = CHR$(AA1) : C$(2*I + 2) = CHR$(BB1)
2460      REM the fol two lines limit screen output to nineteen lines at a
           time SLINE% = the number of <SCREEN LINES> currently displayed
              as output
2480        SLINE% = SLINE% + 1 : IF SLINE% >= 20 THEN 2500 ELSE 2560
2500        SLINE% = 0 : CLS
2520      REM the fol line prints output data column headings
2540       PRINT " Byte #  Byte #  Integer  Integer  ASCII Characters "
2560    NEXT I
2580 REM *************** ZERO B%(matrix) & ReDIM A%(matrix)  ***************
2600       ERASE B%
2620       PRINT : PRINT "Erasing B%(matrix) to conserve memory space."
2640       DIM A%(6600) : PRINT "ReDIM A%(matrix) for X & Y coordinates."
2660 REM -----------------------------------------------------------------
2680 REM *****************  Decoding X and Y Integer Coords  ***************
2700 REM RE-USING A%(matrix) to hold integer graphic X & Y coordinate values
              decoded from disc file and stored in C$(matrix).
2720 REM A%(index) to A%(index + 2299) == X-coordinates storage
2740 REM A%(index + 3300) to A%(6600) == Y-coordinates storage
2760 PRINT : PRINT "Decoding ASCII graphic data in C$(matrix) into integer
              X and Y coordinates."
2780 PRINT : PRINT "Values stored in C$(matrix) are ... "
2800    N = 1 : REM Set initial A%(matrix index) and A%(matrix index + 3300)
2820    X$ = "" : Y$ = "" : REM Set temporary X & Y string values to null.
2840    REM The index JK tracks C$(matrix) elements being printed to screen.
2860    JK = 1 : REM Set secondary index for next loop to initial value.
2880    REM The index J tracks C$(matrix) elements being decoded.
2900        J = 1 : REM Set J-index to initial value of one.
2920 REM ASCII values tested for in fol loop are ASCII(65) = A  ASCII(44) = ,
          and  ASCII(59) = ;
2940 REM The character "A" leads the X & Y coord values; "," seperates the
      X & Y coord values; and ";" seperates the coord value pairs in the
           data string from the Signal Analyzer.
2960      PRINT C$(JK);
2980            IF ASC(C$(J)) <> 65 THEN 3220
3000            J = J + 1
```

```
3020            X$ = X$ + C$(J)
3040            J = J + 1
3060            IF ASC(C$(J)) <> 44 THEN 3020
3080            REM Transfer decoded X coord to A%(matrix).
3100              A%(N) = VAL(X$) : J = J + 1
3120            Y$ = Y$ + C$(J) : J = J + 1
3140            IF ASC(C$(J)) <> 59 THEN 3120
3160            REM Transfer decoded Y coord to A%(matrix INDEX + 3300).
3180              A%(N + 3300) = VAL(Y$) : N = N + 1
3200            X$ = "" : Y$ = "" : REM Reset temporary string values to null.
3220            JK = JK + 1 : J = J + 1
3240      IF 2*(EB%-SB%) - J > 0 THEN 2960 : REM End coordinate loop.
3260 REM ************** Start loop to store XY-coordinates on DISC  **********
3280      CLS : LOCATE 4,1 : SOUND 800,10 : SOUND 760,8
3300      PRINT "Graphic XY-coordinates for ";FILE$;" have been determined. "
3320      PRINT : INPUT "Type  <YES>  to store XY-coordinates... ",ANS$
3340      IF LEFT$(ANS$,1) = "Y" THEN 3380
3360      IF LEFT$(ANS$,1) <> "y" THEN 3620
3380         PRINT : PRINT "What XY-coord filename?"
3400         PRINT:PRINT "File designations must be  XYC  (ie)  FILENAME.XYC "
3420         PRINT : INPUT "Filename.XYC ... ";XYFILE$
3440            IF RIGHT$(XYFILE$,3) = "XYC" THEN 3480
3460            IF RIGHT$(XYFILE$,3) <> "xyc" THEN 3380
3480         PRINT : PRINT "Storing your XY-graphic coordinates... "
3500         OPEN "C:\GPIB-PC\" + XYFILE$ FOR OUTPUT AS #2
3520           FOR J = 1 TO N
3540            PRINT #2, USING "######"; A%(J); A%(J + 3300);
3560           NEXT J
3580             PRINT #2, USING "######"; 0;0;0;0
3600         CLOSE #2
3620 REM ****************  Find XMAX & YMAX & Y3DB DOWN  ******************
3640 REM +++++++++++++++++ NULL C$(matrix) ++++++++++++++++++++++++++++
3660         ERASE C$ : REM Erasing C$(matrix) to conserve memory space.
3680 REM -------------------------------------------------------------------
3700 PRINT : PRINT : PRINT "Identifying XMAX and YMAX plus Y3DB-DOWN."
3720 PRINT : PRINT "Graphic numeric X & Y coordinates are ..."
3740      XMAX% = 0 : YMAX% = 0 : REM Set maximum initial XY coords to zero.
3760        FOR JJ = 1 TO N-1
3780           PRINT "     X(";JJ;") =  ";A%(JJ);SPC(4);
3800           PRINT "Y(";JJ;") =  ";A%(JJ + 3300)
3820           IF A%(JJ + 3300) < YMAX% THEN 3860
3840              YMAX% = A%(JJ + 3300) : XMAX% = A%(JJ) : INDEX = JJ
3860        NEXT JJ
3880           YBTM% = (A%(3300 + N - 2) + A%(3302))/2!
3900           Y3DBD = .7071067 * (YMAX% - YBTM%) + YBTM%
3920 PRINT : PRINT "Graphic file contained ";FE - 3;" total data integers."
3940 PRINT : PRINT "XMAX = ";XMAX%;"  YMAX = ";YMAX%
3960 PRINT "Y(3db down) =";Y3DBD
3980 REM **************  ID W1 and W2  **********************************
4000 PRINT : PRINT "Identifying W1 and W2 at Y(3db down)... "
4020   TOL = 2.5 : REM Set comparison tolerance to two and one-half.
4040 REM Start loop to locate/identify horiz axis values (freq) X1 and X2
      corres to Y(3db down) values to the left & right of peak db-amplitude.
4060   TL = -100! : REM Set test index for next loop to zero.
4080   X1 = 0! : X2 = 0! : REM Set initial 3db-down X-coords to zero.
```

```
4100      FOR KK = 1 TO N-1 : REM Start loop to identify X1 and X2.
4120            PRINT "$";
4140        IF ABS(A%(KK + 3300) - Y3DBD) > TOL THEN 4280
             : REM Check tolerance.
4160           REM Skip setting X1 value if previously found.
4180           IF TL > 0! THEN 4260
4200           X1 = A%(KK) : TL = 10! : REM Set X1 and TL
4220             REM Move to far side of curve amplitude maximum.
4240             KK = INDEX : GOTO 4280
4260           X2 = A%(KK) : GOTO 4320 : REM Set X2 and exit loop.
4280      NEXT KK
4300 REM ****************** SDC Correction Factor Section ****************
4320     PRINT : PRINT
4340       INPUT "What SDC/DC correction factor";CF$
4360       CF = VAL(CF$)
4380     IF CF = 0! THEN 4320
4400     PRINT : PRINT
4420 REM ******************  Calculate SDC & DC  *************************
4440 REM Calculate Specific Damping Capacity (SDC) & Damping Coefficient (DC)
4460   SDC = 200!*3.1415926#*((X2 - X1)/XMAX%)
4480     SDC = SDC/CF
4500   DC = (X2 - X1)/(2! * XMAX%)
4520     DC = DC/CF
4540   PRINT : PRINT "*******************************************************"
4560   PRINT : PRINT "  VALUES FOR ";FILE$;SPC(3);"YMAX(Abs) = ";YMAX%
4580   PRINT : PRINT "  SDC = "; SDC; " % ";SPC(3);"X(0) = ";A%(1)
4600   PRINT : PRINT "  DC  = "; DC ;SPC(3); "XMAX(Abs) = ";A%(N-2)
4620   PRINT : PRINT "-------------------------------------------------------"
4640   PRINT : PRINT " SDC/DC CORRECTION FACTOR OF ";CF
4660   PRINT : PRINT "*******************************************************"
4680 PRINT:INPUT "Type any key + <RETURN> to ReEnter the SDC/DC factor ",ANS$
4700      IF ANS$ <> "" THEN 4320
4720      PRINT : PRINT
4740 REM *****************************************************************
4760 PRINT : INPUT "Type  yes  to examine another disc file... "; ANS$
4780    IF LEFT$(ANS$,1) = "y" THEN 100
4800    IF LEFT$(ANS$,1) = "Y" THEN 100
4820    RUN "MENU.BAS"
4840 END
4860 REM ************ No existing disc file error trap **********************
4880      IF ERR = 53 THEN PRINT "File not found; try again. " : RESUME 340
4900      ON ERROR GOTO 0
4920 REM ****************** KEY (9) INTERRUPT SUBROUTINE ********************
4940       CLS : LOCATE 4, 1 : BEEP : BEEP
4960       PRINT : PRINT "PROGRAM INTERRUPT... "
4980       PRINT : PRINT "Type  <RETURN>  to resume this program section."
5000       PRINT : PRINT "Type  <KK>  to start this program section over."
5020       PRINT : PRINT "Type any other key + <RETURN> to exit to main menu."
5040       PRINT : INPUT ANS$
5060          IF ANS$ = "" THEN RETURN
5080          IF ANS$ = "KK" THEN 100
5100          IF ANS$ = "kk" THEN 100
5120             RUN "menu.bas"
```

```
100 REM ********************* GPIB-PC PROGRAM HEADER ************************
120      CLEAR : REM Clear memory for start or restart.
140 REM ----------  Program Graph Data <GRAPHXYC.BAS>  ------------
160 DIM X%(1000), Y%(1000), YY%(1000)
180 REM X%(xxxxx) is a matrix used to store X-coordinates from the
                   designated disc file in the computer active memory
200 REM Y%(xxxxx) is a matrix used to store Y-coordinates from the
                  designated disc file in the computer active memory
220 REM YY%(xxxxx) is a matrix holding "smoothed" Y-coord values used for
240 REM   calculation of SDC & DC. These coords are displayed on the computer
260 REM   screen for verification that smoothed curve fit is satisfactory.
280 REM   Note that X-coord values remain unchanged in both index and value.
300 REM ******************************************************************
320 REM ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
340   KEY 9, "EXIT" : ON KEY(9) GOSUB 4200 : KEY(9) ON : REM Interrupt trap.
360 REM ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
380 CLS : REM Fol box appears on the screen and defines the program function
400          SOUND 900, 10 : SOUND 840, 9
420 PRINT"**************** Program <GRAPHXYC.BAS> *************************"
440 PRINT"** This program loads the XY-coordinate contents of a designated  **"
460 PRINT"** disc file into the computer, then displays those coordinates   **"
480 PRINT"** graphically on the computer screen.  A second curve is also    **"
500 PRINT"** displayed that is a  SMOOTHED  version of the disc file graph. **"
520 PRINT"** The  SMOOTHED  version is used for SDC & DC calculations, so    **"
540 PRINT"** its visual  FIT  to the disc file graphic data is displayed on **"
560 PRINT"** the computer screen for comparison.  XY-coord disc files used  **"
580 PRINT"** by this program must have been produced by <DAMPCALC.BAS>.      **"
600 PRINT"*****************************************************************"
620 PRINT"** Note: Disc files for this program must have file designations  **"
640 PRINT"** of  <XYC>  (ie)  FILENAME.XYC    Other file types will not     **"
660 PRINT"** load correctly.                                                **"
680 PRINT"*****************************************************************"
700          ON ERROR GOTO 4160 : REM No existing disc file error trap.
720 PRINT : INPUT "Type  <RETURN>  to continue... ",ANS$
740 CLS : FILES "*.xyc"
760 PRINT : PRINT "What file name contains your XY-coord data?"
780 PRINT "File designation must be  xyc  (ie);  filename.xyc  "
800 PRINT : INPUT "Filename.xyc ...";FILE$
820        IF RIGHT$(FILE$,3) = "XYC" THEN 880
840        IF RIGHT$(FILE$,3) <> "xyc" THEN 380
860 REM ******* DISC FILE LOADING SECTION ***********************************
880 CLS : PRINT "Loading the contents of "; FILE$; " into the computer... "
900 PRINT : PRINT "Disc file contents are ... Please wait... "
920 OPEN "c:\gpib-pc\" + FILE$ FOR INPUT AS #1
940    FOR I = 1 TO 6000
960       IF EOF(1) THEN 1120
980      INPUT#1, X%(I), Y%(I)
1000      IF X%(I) + Y%(I) = 0 THEN 1120
1020      PRINT X%(I), Y%(I);
1040      X = CSRLIN : IF X <= 23 THEN 1080
1060         CLS: LOCATE 4,1 : PRINT "Loading contents of ";FILE$
1080    NEXT I
1100 REM FE = the index <File End> of the last graphic coordinate.
1120     FE = I : CLOSE #1
1140 CLS:PRINT : PRINT "Disc file ";FILE$; " is loaded into active memory ... "
```

```
1160 REM ************************************************************************
1180 REM ********* SECTION TO LOCATE XMAX% & YMAX% & THEIR INDEX **************
1200 PRINT : PRINT : PRINT "Identifying XMAX and YMAX."
1220 PRINT : PRINT "Graphic numeric X & Y coordinates are ..."
1240      XMAX% = 0 : YMAX% = 0 : REM Set maximum initial XY coords to zero.
1260    FOR JJ = 1 TO FE-1
1280      PRINT "      ";
1300      PRINT "X(";JJ;") =  ";X%(JJ);SPC(4);
1320      PRINT "Y(";JJ;") =  ";Y%(JJ)
1340         IF Y%(JJ) < YMAX% THEN 1400
1360            YMAX% = Y%(JJ) : XMAX% = X%(JJ)
1380              INDEX = JJ
1400    NEXT JJ
1420 REM ******************** CURVE SMOOTHING SECTION *************************
1440 REM The smoothing routine calculates the average Y-coord increase over
1460 REM  the left half of the maxima curve (LYI == left Y-coord increment)
1480 REM  and the right half of the maxima curve (RYI == right Y-coord incr.)
1500 REM  Then the program adds proportionate amounts of the LYI to the Y-coord
1520 REM  value at the left end (X minimum, Y minimum -- left) of the curve,
1540 REM  continuing to the maximum Y-coord value (marked by INDEX).  The
1560 REM  right half of the curve is determined in a similar manner. However,
1580 REM  the RYI is NEGATIVE, so it is subtacted in proportionate amounts
1600 REM  from the curve maximum Y-coord value and preceeding to the right
1620 REM  end of the curve (X minimum, Y minimum -- right).  Note that the
1640 REM  X-coord values are NOT changed in index nor magnitude.
1660 PRINT : PRINT "Smoothing the curve for ";FILE$;" ....... "
1680     FOR KK = 1 TO INDEX
1700       REM LYI == left half of graph Y-coords average increment
1720       LYI = LYI + (Y%(KK+1) - Y%(KK))
1740     NEXT KK
1760       LYI = LYI/INDEX : REM LYI now set as equal Y-coord increment
              per coord
1780     FOR MM = INDEX TO FE-2
1800       REM RYI == right half of graph Y-coords average increment
1820       RYI = RYI + (Y%(MM+1) - Y%(MM))
1840     NEXT MM
1860       RYI = RYI/(FE-1-INDEX) : REM RYI now set as equal Y-coord increment
1880     FOR NN = 1 TO INDEX
1900       REM A smoothing factor of 1.05 was used for the left half of the
1920       REM  curve.  This factor can be changed by the programmer to suit
1940       REM   his curve fitting preferences.
1960       REM Smooth left half of graph Y-coords (smoothing factor 1.05).
1980       YY%(NN) = Y%(1) + ((NN-1) * LYI) * 1.05 * (Y%(NN)/YMAX%)^.5
2000     NEXT NN
2020          XXX = 1!
2040     FOR OO = INDEX+1 TO FE-1
2060       REM A smoothing factor of 1.20 was used for the right half of the
2080       REM  curve.  This factor can be changed by the programmer to suit
2100       REM   his curve fitting preferences.
2120       REM Smooth right half of graph Y-coords (smoothing factor 1.2).
2140       YY%(OO) = Y%(INDEX) + ( XXX * RYI) * 1.2 * (Y%(FE-1)/Y%(OO))^.5
2160          XXX = XXX + 1!
2180     NEXT OO
2200 REM *************  Calculate Y3db down  (Half-power Y-coord)  ************
2220              YBTM% = (Y%(2) + Y%(FE -2))/2!
```

```
2240                Y3DBD = .7071067 * (YMAX% - YBTM%) + YBTM%
2260 PRINT : PRINT "Graphic file contained ";FE - 1;" total XY-coordinates."
2280 PRINT: PRINT "XMAX = ";XMAX%;"  YMAX = ";YMAX%;"
                                      INDEX OF X&Y MAX = ";INDEX
2300 PRINT "Y(3db down) =";Y3DBD
2320 REM *********** SECTION TO IDENTIFY W1 & W2 ******************************
2340 PRINT : PRINT "Identifying W1 and W2 at Y(3db down)... "
2360   TOL = 2.5 : REM Set tolerance for selection of W1 and W2.
2380 REM Start loops to locate/identify horiz axis values (freq) W1 and W2
        corres to Y(3db down) values to the left & right of peak
            db-amplitude.
2400  W1 = 0! : W2 = 0! : REM Set W1 & W2 initially to zero.
2420    REM Start loop for finding W1.
2440      FOR KK = INDEX TO 1 STEP -1
2460           PRINT "$";
2480        IF ABS(YY%(KK) - Y3DBD) > TOL THEN 2540 : REM Identity tolerance.
2500           W1 = X%(KK) : REM Set W1 to identified value.
2520              PRINT : PRINT : PRINT "W1 FOUND ..." : GOTO 2580
2540      NEXT KK
2560    REM Start loop for finding W2.
2580      FOR KL = INDEX TO FE - 1
2600           PRINT "$";
2620        IF ABS(YY%(KL) - Y3DBD) > TOL THEN 2680 : REM Identity tolerance.
2640           W2 = X%(KL) : REM Set W2 to identified value.
2660              PRINT : PRINT : PRINT "W2 FOUND ..." : GOTO 2700
2680      NEXT KL
2700   PRINT : PRINT "W1 = ";W1;SPC(2);"W2 = ";W2 : PRINT
2720 REM ********* SECTION TO CALCULATE SDC & LD ******************************
2740   PRINT : INPUT "What SDC/DC correction factor";CF$
2760    CF = VAL(CF$)
2780    IF CF = 0! THEN 2740
2800 REM Calculate Specific Damping Capacity (SDC) & Damping Coefficient (DC)
2820   SDC = 200!*3.1415926#*((W2 - W1)/XMAX%)
2840     SDC = SDC/CF
2860   DC = (W2 - W1)/(2! * XMAX%)
2880     DC = DC/CF
2900  CLS : PRINT : PRINT : REM Clear screen and position results display.
2920      PRINT : PRINT "*******************************************"
2940      PRINT : PRINT "      RESULTS FOR "; FILE$
2960      PRINT : PRINT "      SDC = "; SDC; " Percent "
2980      PRINT : PRINT "      DC  = "; DC
3000      PRINT : PRINT "  PLOTTER COORD YMAX(Abs) = ";YMAX%
3020      PRINT : PRINT "  X(0) = "; X%(1);SPC(2);"X(MAX) = ";X%(FE-2)
3040      PRINT : PRINT "  SDC/DC CORRECTION FACTOR IS ";CF
3060      PRINT : PRINT "*******************************************"
3080 PRINT : INPUT "Type any key + <RETURN> to ReEnter the SDC/DC factor ",ANS$
3100    IF ANS$ <> "" THEN 2720
3120    PRINT : PRINT
3140    INPUT "Type  <RETURN>  for graph of file data.",ANS$
3160 REM ************* SECTION TO PLOT GRAPHICS ***************************
3180 REM XFCTR & YFCTR permit moving the screen trace about the screen X-Y
      coordinate field.  A positive XFCTR moves trace to the right.
       A positive YFCTR moves the trace down.
3200      XFCTR = 0! : YFCTR = 0!
3220      CLS : SCREEN 2
```

```
3240  LINE (0,0) - (639,199),1,B
3260  PSET (0, 199), 1
3280    FOR MM = 1 TO FE-1
3300      X1 =  (X%(MM) * 638!)/X%(FE-1) + XFCTR
3320      Y1 = 198! - (Y%(MM) * 198!)/YMAX% + YFCTR
3340      LINE - (X1, Y1)
3360    NEXT MM
3380  PSET (0, 199), 1
3400    FOR MN = 1 TO FE-1
3420      X1 =  (X%(MN) * 638!)/X%(FE-1) + XFCTR
3440      Y1 = 198! - (YY%(MN) * 198!)/YMAX% + YFCTR
3460      LINE - (X1, Y1)
3480    NEXT MN
3500 REM ************ Mark W1 and W2 Positions with Vertical Lines ************
3520   LINE (W1*638!/X%(FE-1)+XFCTR, YMAX%/10!) - (W1*638!/X%(FE-1)+XFCTR, 198)
3540   LINE (W1*636!/X%(FE-1)+XFCTR, YMAX%/10!) - (W1*636!/X%(FE-1)+XFCTR, 198)
3560   LINE (W1*640!/X%(FE-1)+XFCTR, YMAX%/10!) - (W1*640!/X%(FE-1)+XFCTR, 198)
3580   LINE (W2*638!/X%(FE-1)+XFCTR, YMAX%/10!) - (W2*638!/X%(FE-1)+XFCTR, 198)
3600   LINE (W2*636!/X%(FE-1)+XFCTR, YMAX%/10!) - (W2*636!/X%(FE-1)+XFCTR, 198)
3620   LINE (W2*640!/X%(FE-1)+XFCTR, YMAX%/10!) - (W2*640!/X%(FE-1)+XFCTR, 198)
3640 REM *********************** Plot Grid Lines *****************************
3660   LINE (0,20) - (638,20) : LINE (0,40) - (638,40)
3680   LINE (0,60) - (638,60) : LINE (0,80) - (638,80)
3700   LINE (0,100) - (638,100) : LINE (0,120) - (638,120)
3720   LINE (0,140) - (638,140) : LINE (0,160) - (638,160)
3740   LINE (0,180) - (638,180)
3760      LINE (64,0) - (64,198) : LINE (128,0) - (128,198)
3780      LINE (192,0) - (192,198) : LINE (256,0) - (256,198)
3800      LINE (320,0) - (320,198) : LINE (384,0) - (384,198)
3820      LINE (448,0) - (448,198) : LINE (512,0) - (512,198)
3840      LINE (576,0) - (576,198)
3860 REM -----------------------------------------------------------------
3880 PRINT "SCREEN TRACE OF ";FILE$;" SIGNAL ANALYZER DISC FILE"
3900 PRINT "A positive XFCTR moves trace to the right.  A positive YFCTR
                                                                   moves "
3920 PRINT "screen trace down.  <XFCTR + YFCTR = 0> exits the graph mode ... "
3940  INPUT "WHAT XFCTR = ";XFCTR
3960  INPUT "WHAT YFCTR = ";YFCTR
3980    IF XFCTR + YFCTR <> 0! THEN 3220
4000 REM *************** PROGRAM ENDING ************************************
4020      SCREEN 0 : REM Set the text screen.
4040 PRINT : INPUT "Type  yes  to examine another disc file... "; ANS$
4060 IF LEFT$(ANS$,1) = "y" THEN 100
4080 IF LEFT$(ANS$,1) = "Y" THEN 100
4100    RUN "MENU.BAS"
4120 END
4140 REM ************* No existing disc file error trap ******************
4160     IF ERR = 53 THEN PRINT "File not found; try again. " : RESUME 400
4180      ON ERROR GOTO 0
4200 REM ********************* INTERRUPT TRAP SUBROUTINE ****************
4220      BEEP : BEEP : CLS
4240      LOCATE 4,1 : PRINT "PROGRAM INTERRUPTED..."
4260      PRINT : PRINT "TYPE  <RETURN>  TO RESUME PROGRAM EXECUTION."
4280      PRINT : PRINT "TYPE  KK  TO SELECT ANOTHER DATA FILE."
4300      PRINT : PRINT "TYPE ANY OTHER KEY + <RETURN> TO EXIT TO MAIN MENU."
```

```
4320      INPUT ; ANS$
4340      IF ANS$ = "" THEN RETURN
4360      IF ANS$ = "KK" THEN 4000
4380      IF ANS$ = "kk" THEN 4000
4400      RUN "MENU.BAS"
```

APPENDIX F

FIGURES

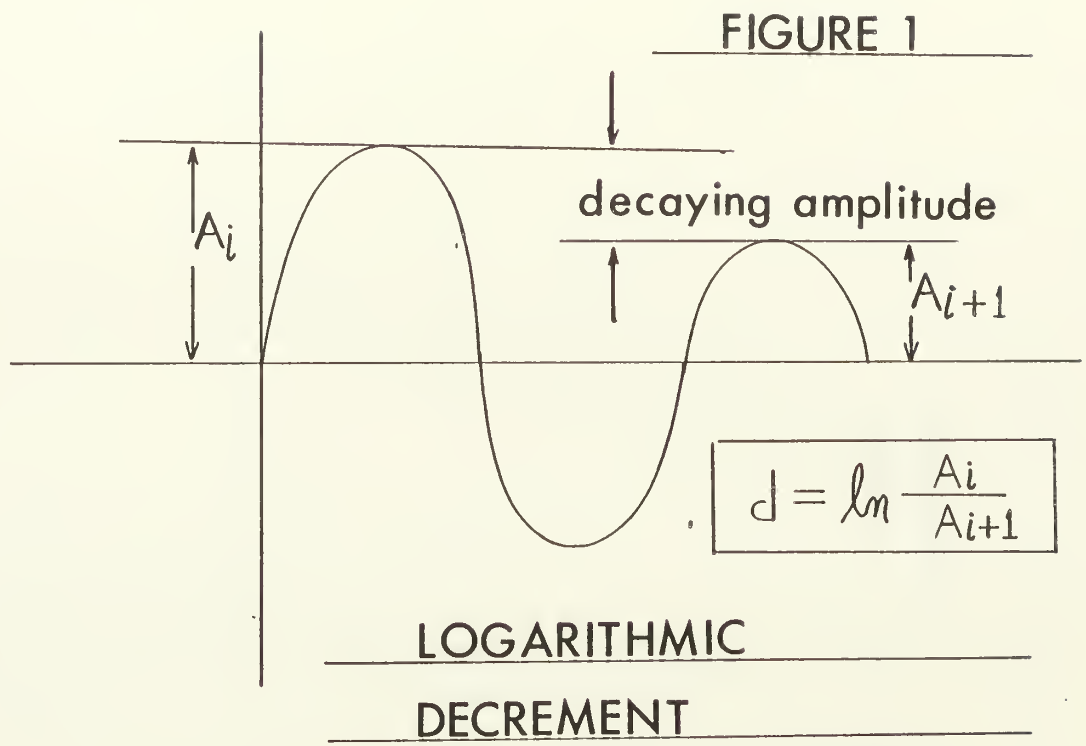
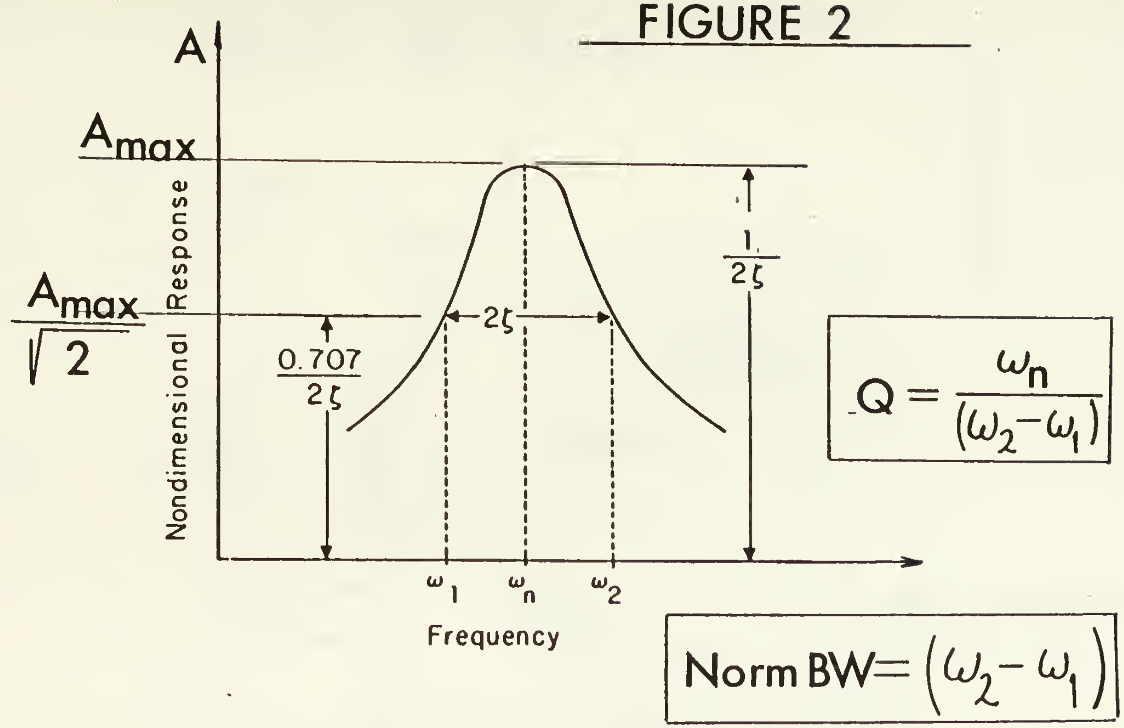


FIGURE 1 LOGARITHMIC DECREMENT

FIGURE 2



QUALITY FACTOR & NORM.
BANDWIDTH

FIGURE 2 QUALITY FACTOR & NORMALIZED BANDWIDTH

FIGURE 3

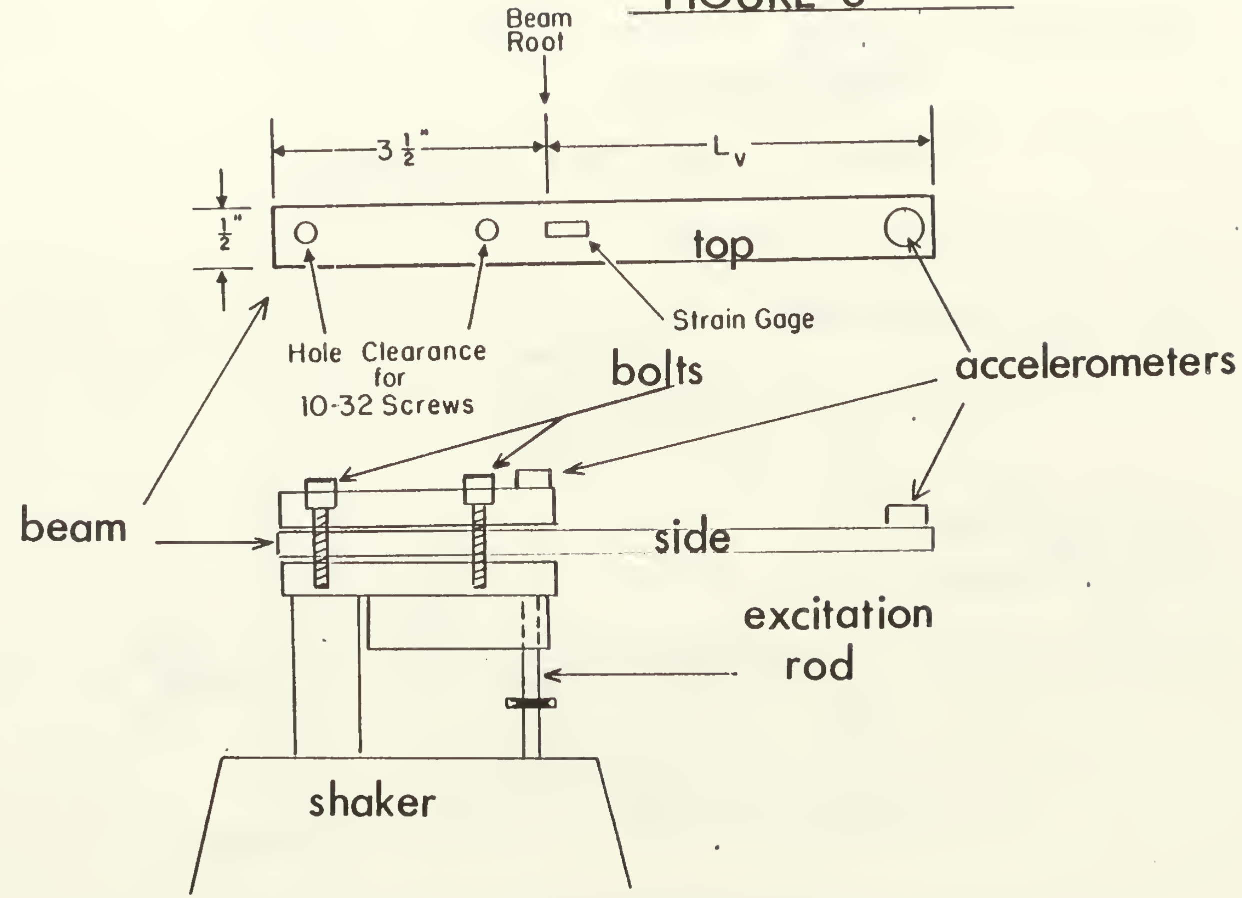


FIGURE 3 BEAM SPECIMEN & ITS POSITION ABOVE THE SHAKER APPARATUS

Cr-Fe (Chromium-Iron)

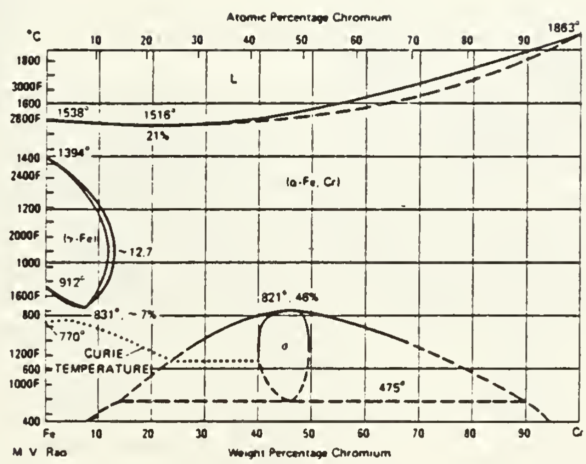


Figure 4 Phase Diagram of Iron-Chromium Binary System

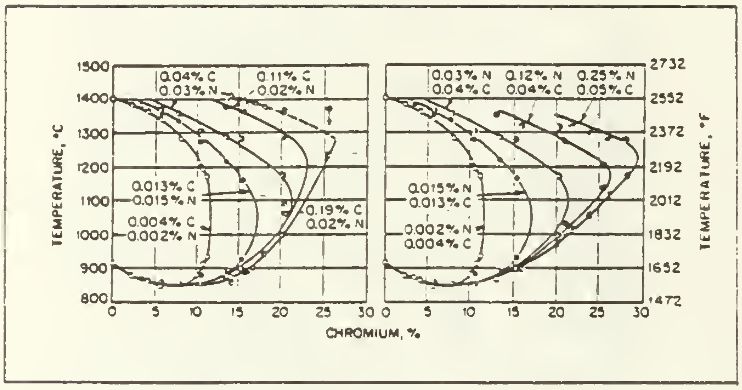


Figure 5 Shifting of the boundary line $(\gamma+\alpha)/\alpha$ in the Fe-Cr system through the addition of carbon and nitrogen

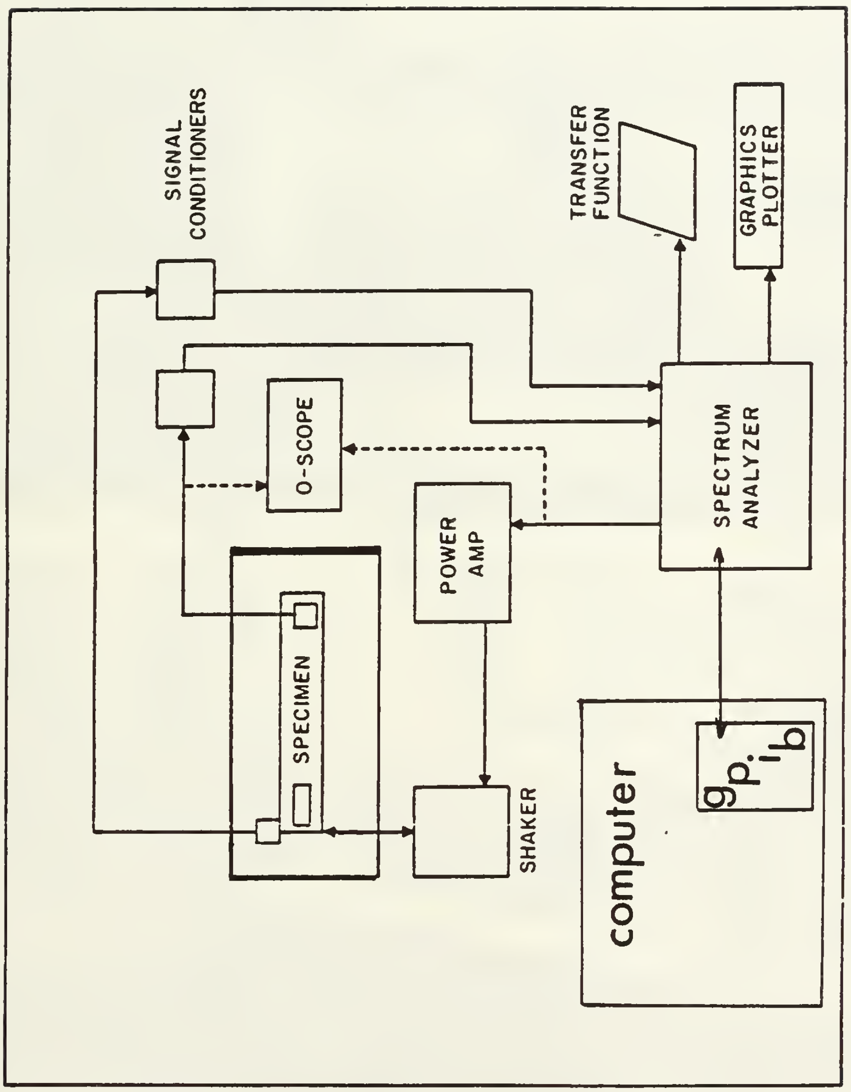

SIGNAL
CONDITIONERS
TRANSFER
FUNCTION
GRAPHICS
PLOTTER
O-SCOPE
SPECTRUM
ANALYZER
POWER
AMP
SPECIMEN
SHAKER
computer
g_{p_i}, i_b

FIGURE 6

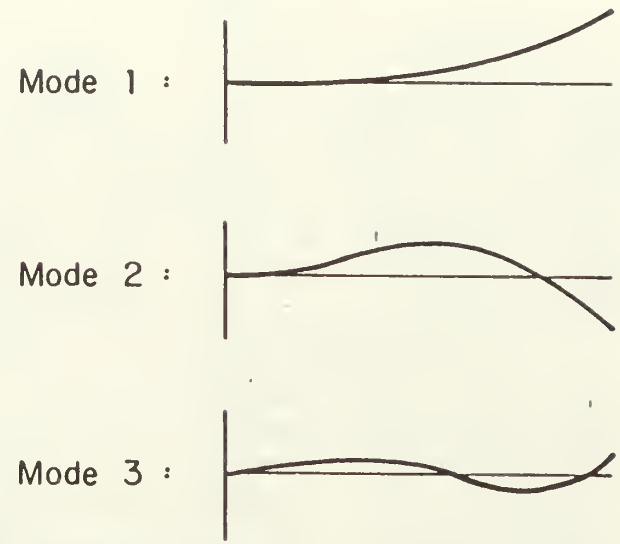
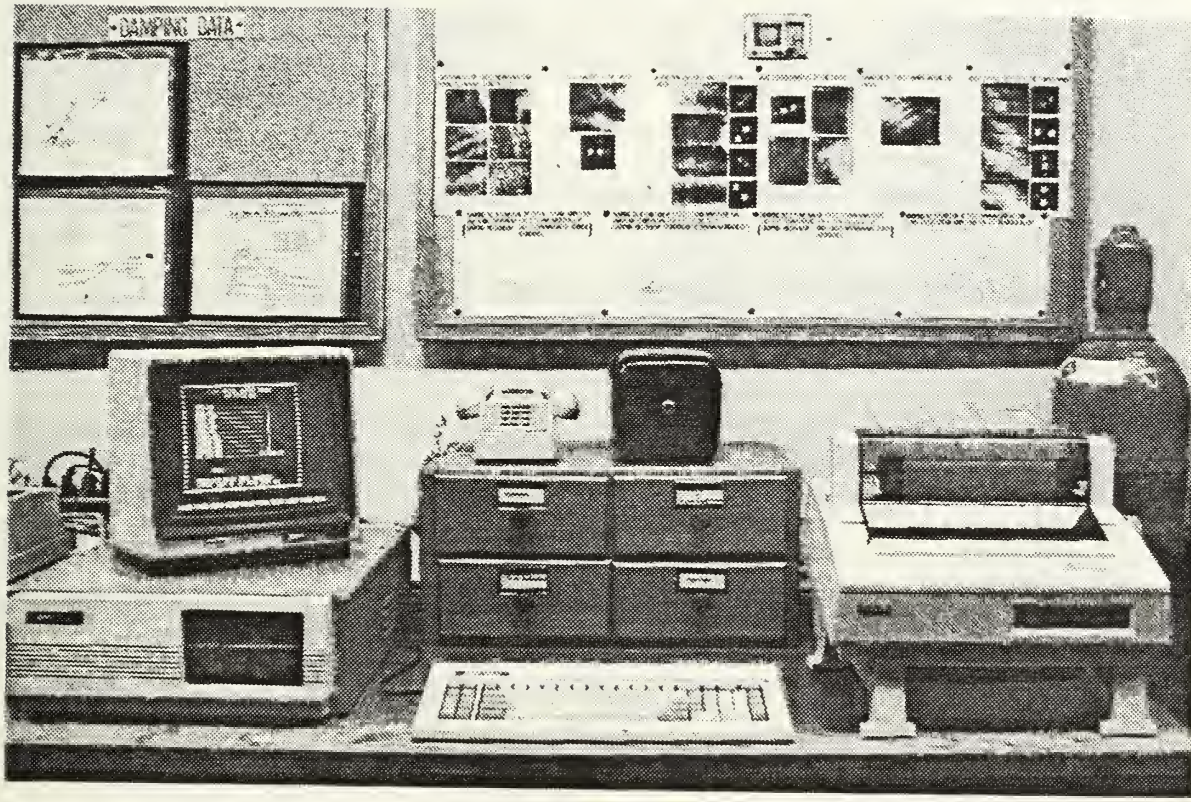


FIGURE 7
CANTILEVER BEAM MODE SHAPES



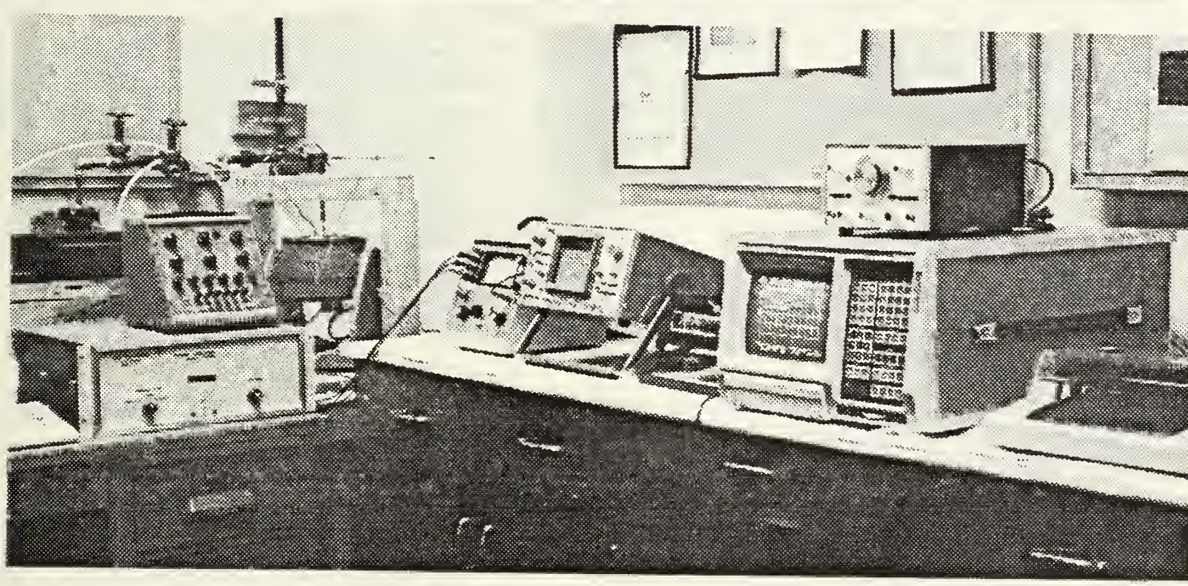


FIGURE 8 RESEARCH EQUIPMENT

LIST OF REFERENCES

1. O'Toole, J.F. and Perkins, J., "Damping Behavior of an Fe-Cr-Mo Alloy: Strain Dependence and Heat Treatment Effects," M.E. Thesis, U.S. Naval Postgraduate School, Monterey, Ca., December 1986.

2. Schetky, L.M. and Perkins, J., "The 'Quiet' Alloys," Machine Design, pp. 202-206, 6 April 1978.

3. Naval Research Laboratory Internal Report, "Acoustic Damping Materials, A Review", by R.A. Meussner and B.B. Rath, November 1982.

4. Escue, W.D., "Characterization of the Corrosion Behavior of High Damping Alloys in Seawater," M.E. Thesis, U.S. Naval Postgraduate School, Monterey, Ca., June 1987.

5. de Batist, R., "High Damping Materials: Mechanisms and Applications," Journal de Physique, Vol. 44, pp. C9-39 - C9-50, December 1983.

6. Granato, A. and Lucke, K., "Theory of Mechanical Damping Due to Dislocations," Journal of Applied Physics, Vol. 17, No. 6, pp. 583-593, June 1956.

7. Dew, D.D., "Strain Dependent Damping Characteristics of a High Damping Manganese-Copper Alloy," M.E. Thesis, Naval Postgraduate School, Monterey, California, September 1986.

8. Reskusich, J., "Cyclic Strain Amplitude and Heat Treatment Effects on the High Damping Behavior of Incramute Alloy Under Random Vibration Loading in the 50-1000 Hz Frequency Range," M.E. Thesis, Naval Postgraduate School, Monterey, California, September 1986.

8a. Ferguson, D., LCDR, USN, M.E. Thesis, Naval Postgraduate School, Monterey, California, Estimated Publication, October 1987.

9. Cochardt, A.W., "The Origin of Damping in High-Strength Ferromagnetic Alloys," Transactions of the ASME, Vol. 75, pp. 196-200, 1953.

10. Schilling, J.W. and Houze, G.L., "Magnetic Properties and Domain Structure in Grain-Oriented 3% Si-Fe," IEEE Transactions on Magnetics, Vol. MAG-10, No. 2, pp. 195-222, June 1974.

11. Willertz, L.E., "Magnetomechanical Damping Properties of AISI 403 Stainless Steel with Applied Static Torsional and Axial Stresses," Journal of Testing and Evaluation, JTEVA, Vol. 2, No. 6, pp. 478-482, November 1974.

12. Suzuki, K., Fijita, T., and Haseb, M., "Damping Capacity and Mechanical Properties of Sintered Fe-Cr-Mo High Damping Alloy," Powder Metallurgy, No. 4, pp. 205-211, 1977.

13. Masumoto, H., Sawaya, S., and Hinai, M., "On the Damping Capacity of Fe-Cr Alloys," Journal of Japan Institute of Metals, (in Japanese), 43, pp. 409-413, 1970.

14. Masumoto, H., Sawaya, S., and Hinai, M., "Damping Characteristics of 'Gentalloy' in the Fe-Mo System," Transactions of the Japan Institute of Metals, Vol. 22, No. 9, pp. 607-613, 1981.

15. Masumoto, H., Sawaya, S., and Hinai, M., "Damping Capacity and Pitting Corrosion Resistance of Fe-Mo-Cr Alloys," Transactions of the Japan Institute of Metals, Vol. 25, No. 12, pp. 891-899, 1984.

16. Schneider, W., Schrey, P., Hausch, G., and Torok, E., "Damping Capacity of Fe-Cr and Fe-Cr Based High Damping Alloys," Journal De Physique, Colloque C5, Supplement au n 10, Tome 42, pp. C5-C35, October 1981.

17. Naval Underwater Systems Center Report TD 6927, "A Technical Update on Damped Vacrosil-010", by R.G. Kasper, 19 September 1983.

18. Kawabe, H. and Kuwahara, K., "Effects of Material Damping on the Suppression of Mechanical Vibrations," Bulletin of the JSPE, Vol. 14, No. 1, pp. 31-36, March 1980.

19. Kawabe, H. and Kuwahara, K., "A Consideration of the Strain Amplitude-Dependent Damping and Modulus in Ferromagnetic Metals," Transactions of the Japan Institute of Metals, Vol. 22, No. 5, pp. 301-308, 1981.

20. Amano, K., Sahashi, M., Tokoro, H., and Nakagawa, M., "High Damping Characteristics of Fe-Cr-Al Alloys Associated with their Magnetic Properties," Proceedings of the 6th International Conference on Internal Friction and Ultrasonic Attenuation in Solids, Ed. R.R. Hasiguti and N. Mikoshiba, University of Tokyo Press, pp. 763-767, 4-7 July 1977.

21. Kawabe, H. and Kuwahara, K., "Damping - and Modulus-Measurement of High-Damping Metal 'SIA' at Low Frequencies," Proceedings of the 6th International Conference on Internal Friction and Ultrasonic Attenuation in Solids, Ed. R.R. Hasiguti and N. Mikoshiba, University of Tokyo Press, pp. 781-785, 4-7 July 1977.

22. Bert, C.W., "Material Damping: An Introductory Review of Mathematical Models, Measures, and Experimental Techniques," Journal of Sound and Vibration, Vol. 29, No. 2, pp. 129-153, 1973.

23. Thomson, W.T., Theory of Vibrations with Applications, Prentice-Hall, Inc., 1981.

24. Reed-Hill, R.E., Physical Metallurgical Principles, Brooks/Cole Engineering Div., 1973.

25. Bolt, Beranek, and Newman, Inc., Cambridge, Massachusetts, Operations Manual for the Bolt, Beranek, and Newman, Inc., Resonant Dwell Apparatus, January 1973.

26. Kaufman, L., Kulin, S.A., and Neshe, P., "Internal Vibration Absorption in Potential Structural Materials," Shape Memory Effects in Alloys, Jeff Perkins, ed., Plenum Press, pp. 547-561, 1975.

27. Cronauer, J. and Perkins, J., "A Comparison of High Damping Shape Memory Alloys with Cu-Mn-Based and Fe-Cr-Based Alloys," M.E. Thesis, U.S. Naval Postgraduate School, Monterey, Ca., June 1987.

27A. DeGauque, J., Astie, B., and Kubin, L.P., "Evidence for the Interaction Between Magnetic Domain Walls and Dislocations in High-Purity Iron from Magnetomechanical Damping Experiments," Phys. Stat. Sol., Part A, Vol. 45, pp. 493-501, 1978.

28. Askeland, D.R., The Science and Engineering of Materials, Wadsworth, Inc., 1984.

29. Fuller, H.W. and Hale, M.E., Journal of Applied Physics 31, P. 238, 1960.

30. Amelinckx, S., The Direct Observation of Dislocations, Academic Press, p. 399 - 405, 1964.

31. Wolfenden, A., "Damping in Turbine Blade Alloys," Journal of Testing and Evaluation, JTEVA, Vol. 10, No. 1, pp. 17-20, January 1982.

32. Peckner, D. and Bernstein, I.M., Handbook of Stainless Steels, McGraw-Hill, Inc., 1977.

33. Grigor'ev, S.B. and Kudryashova, L.K., "Magnetoelastic Attenuation due to Rotation of Magnetization," Soviet Phys. Solid State, Vol. 17, No. 1, pp. 99-102, July 1975.

34. Metals Handbook, 8th ed., Vol. 8, American Society for Metals, 1973.

35. Z-150 Personal Computer Desktop System, Operations Manual, Zenith Data Systems Corporation, St. Joseph, Michigan, 1985.

36. Everdisk Internal/External Hard Disk Subsystem (XT Compatible), Owner's Manual and Reference Guide, Everex Systems, Inc., Fremont, Ca., June 1986.

37. GPIB-PC User Manual for the IBM Personal Computer and Compatibles, National Instruments, Austin, Texas, April 1986.

38. Interfacing and Programming Manual, HP-7470A Graphics Plotter, Hewlett-Packard Company, San Diego, Ca., 1984.

39. SD380Z Signal Analyzer Operator's Manual, Scientific-Atlanta Spectral Dynamics Division, San Diego, Ca.

40. Model PM25 Vibration Exciter, Instruction Manual, MB Dynamics, Cleveland, Ohio.

41. Model 2125 MB Power Amplifier, Instruction Manual, MB Dynamics, Cleveland, Ohio.

42. ENDEVCO Piezoelectric Accelerometers, Instruction Manual, ENDEVCO, San Juan Capistrano, Ca., October 1979.

43. ENDEVCO Models 4416 and 4416A Signal Conditioner, Instruction Manual, ENDEVCO, San Juan Capistrano, Ca., November 1980 (Rev)

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	1
3.	Department Chairman, Code 69Hy Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	1
4.	Professor A. J. Perkins, Code 69Ps Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	12
5.	Professor Y. S. Shin, Code 69SG Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5000	1
6.	CDR G. R. Patch, USN PRODUCTION OFFICER (C-300) U. S. Naval Ship Repair Facility Guam FPO San Francisco 96637	2
7.	Mr. Robert Hardy, Code 2803 David W. Taylor Naval Ship R & D Center Annapolis, Maryland 21402	5
8.	Ms. Cathy Wong, Code 2812 David W. Taylor Naval Ship R & D Center Annapolis, Maryland 21402	2
9.	CDR D. P. Mahoney, USN Naval Engineering Curricular Officer (C-34) Naval Postgraduate School Monterey, California 93943-5000	1
10.	Commanding Officer U. S. Naval Ship Repair Facility Guam FPO San Francisco 96637	1

Thesis
P2592 Patch
c.1 Investigation of two ferromagnetic damping materials in conjunction with initial development of a signal analyzer interface program.